Robotics and Automation in the Maritime Industries

Edited by

J. Aranda
P. Gonzalez de Santos
J. M. de la Cruz

Robotics and Automation in the Maritime Industries

Robotics and Automation in the Maritime Industries

Edited by

J. Aranda
Department of Computer Science and Automatic Control
UNED, Madrid

P. Gonzalez de Santos
Industrial Automation Institute
Spanish National Council for Research - CSIC

J. M. de la Cruz
Faculty of Physics
Complutense University of Madrid

First Published 2006

ISBN-13: 978-84-611-3915-6
ISBN-10: 84-611-3915-1

Printed by Producción Gráfica Multimedia, PGM, Madrid, Spain

ABOUT THE EDITORS

Joaquin Aranda received his Licentiate degree form the Complutense University of Madrid, and the Ph.D. degree from the UNED (Universidad Nacional de Educación a Distancia). He served as Teaching Assistant of Department of Computer Science and Automatic Control, Complutense University of Madrid, and as Assistant Professor, Associated Professor and Senior Professor at the UNED. He was deputy director of the Computer Science University School of UNED, and director of the Computer Science High School. He is author or co-author of more than 70 publications including book chapters, papers in journals and conference proceedings. His scientific activities cover various aspects within the control engineering field: robust control, computer control, modelling and simulation of continuous processes, and application to high speed craft, airplane and robotics. He is working in serveral research projects relating to these topics.

Pablo Gonzalez de Santos is a research scientist at the Spanish National Council for Research (CSIC). He received his Ph.D. degree from the University of Valladolid, Spain. Since 1981, he has been involved actively in the design and development of industrial robots and also in special robotic systems. His work during the last fifteen years has been focused to walking machines. He worked on the AMBLER project as a visiting scientist at the Robotics Institute of Carnegie Mellon University. Since then, he has been leading the development of several walking robots such as the RIMHO robot designed for intervention on hazardous environments, the ROWER walking machine developed for welding tasks in ship erection processes and the SILO4 and SILO6 walking robots. He has also participated in the development of other robotic system for maritime industries such as the REST climbing robot intended for welding in the vertical walls of ship hulls and the AURORA system developed for cleaning the adherences of underwater ship hulls.

Jesus Manuel de la Cruz received his Ph.D. degree in Physics from the Complutense University of Madrid in 1984. He has been Associated Professor in the area of Automatic and System Engineering, Faculty of Science (UNED), during the period 1986-1990. He is Full Professor at the Faculty of Physics (Complutense University of Madrid), since 1992. He has been Director of the Departmental Section of Informatics and Automatic at the Faculty of Physics (Complutense University of Madrid), 1993-1997, and Director of the Departament of Computer Architecture and Automatic, 1997-2001. He has participated and managed in many research topics, like those related with: High Speed Ships (Bazan), Robust Control for Aeronautics (EC Garteur FM AG08), Optimization (Repsol-YPF, Enagas, COFARES), Decision making (EADS-CASA), Modelling and Simulation (Indra Sistemas). Professor de la Cruz has directed 10 Ph.D Theses in the Area of Automatic Control, and published over 25 papers in international journals. He is co-author of three books. His present research is focused on control system, optimization and statistical learning. He is a member of the IEEE and chairman of the IEEE - OES (Ocean Engineering Society) Spanish Chapter.

CONTENTS

PREFACE

AUTOMAR Thematic Network (CICYT - DPI202-10620-E and DPI 2004-22181-E) was organised with the idea of putting together all the Spanish research groups and companies involved somehow in the maritime sector in order to increase co-operation level and knowledge dissemination. The AUTOMAR partners are the following Institutions:

Universidad Complutense de Madrid
Departamento de Arquitectura de Ordenadores y Automática
Facultad de Ciencias Físicas
Ciudad Universitaria
28040-Madrid. Spain

Universidad Nacional de Educación a Distancia (UNED)
Departamento de Informática y Automática
Facultad de Ciencias
C) Senda del Rey 9
28040-Madrid. Spain

Universidad de Cantabria
Departamento de Tecnología Electrónica, Ingeniería de Sistemas y Automática
Grupo de Informática y Automática
ETSIIT
Avenida de los Castros s/n
39005-Santander. Spain

Universidade da Coruña
E.S. Marina Civil
Paseo de Ronda 51
15011-A Coruña. Spain

Consejo Superior de Investigaciones Científicas
Instituto de Automática Industrial
Carretera de Campo Real km 0.200
La Poveda. Arganda del Rey
28500-Madrid. Spain

Universidad Politécnica de Madrid
Departamento de Automática, Ingeniería Electrónica e Informática Industrial.
División de Ingeniería de Sistemas y Automática
E.T.S. de Ingenieros Industriales
C) José Gutiérrez Abascal 2
28006-Madrid. Spain

Departamento de Arquitectura y Construcción Navales
Departamento de Sistemas Oceánicos y Navales (Laboratorio de Electrotecnica, Electrónica y Sistemas)
E.T.S. de Ingenieros Navales
Avda Arco de la Victoria s/n
28020-Madrid. Spain

Universidad Politécnica de Cataluña
Departamento de Ingeniería de Sistemas, Automática e Informática Industrial
Edifici U - Campus Sud
c/ Pau Gargallo, 5
08028 Barcelona. Spain

Universidad de Girona
Grupo de Visión por Computador y Robótica
Escuela Politécnica Superior
Campus Montilivi, Edificio Politécnica-4
17003 Girona. Spain

U. de les Illes Balears
Grupo de Sistemas, Robótica y Visión.
Departamento de Ciencias Matemáticas e Informática
Ctra. Valldemossa. Km 7,5.
07122 Palma de Mallorca. Spain

Universidad de Cádiz
Facultad de Ciencias Náuticas
Polígono Río San Pedro s/n
11510 Puerto Real. Cádiz. Spain

Canal de Experiencias Hidrodinámicas de El Pardo
Carretera de la Sierra s/n
El Pardo-Madrid
28048 Spain

AUTOMAR was aiming to foster Spanish research and innovation activity in the maritime industrial sector in order to strengthen their role in Europe. During the six years of the AUTOMAR Network many activities have been carried out, and all the partners have actively contributed. This book, prepared to be presented at the last AUTOMAR meeting, collects in its 14 chapters excellent contributions that show the level of scientific knowledge in this field reached by the Spanish RTD, both at the Universities and at Research Institutions. We would like to thank all of authors and the participants Institutions for their contributions and help. Special thanks are for the Ministry of Education and Science for the AUTOMAR network funding under projects mentioned above.

Joaquin Aranda
Pablo Gonzalez de Santos
Jesus M. de la Cruz

November 2006

CHAPTER 1

Control problems in marine vehicles: Some experiences in stabilization and tracking control

J. ARANDA[1], J.M. DE LA CRUZ[2], R. MUÑOZ-MANSILLA[1], D. CHAOS[1], J.M. DIAZ[1], S. DORMIDO-CANTO[1] and J.M. RIOLA[3]

[1] Departamento de Informática y Automática, UNED, Madrid.
[2] Departamento de Arquitectura de Computadores y Automática. Universidad Complutense de Madrid.
[3] Escuela Técnica Superior de Ingenieros Navales. U. Politécnica de Madrid. jaranda@dia.uned.es

Control problems of underactuated marine systems motivate the development of new control design methodology. Control design of tracking, point stabilization, path following for marine vehicles, or Dynamic Positioning for offshore systems is example of these types of problems. In this chapter, an overview about different problems studied in our laboratory are showed.

1 Introduction

In marine vehicles, we have discerning between guidance and control. The guidance term correspond to the action of determining the course, attitude and speed of the vehicle, relative to some reference frame, usually the earth, to be followed by the vehicle. The control term correspond to the development and application to a vehicle appropriate forces and moments for operating point control, tracking and stabilization. This involves designing the feedforward and feedback control laws.

Furthermore, there are a lot of marine vehicles kinds that are underactuated, i.e., systems with a smaller number of control inputs than the number of independent generalized coordinates.

One of the difficulties encountered in the stabilization and tracking of underactuated vehicles is that classical nonlinear techniques in nonlinear control theory like feedback linearization are not applicable because these systems are not fully feedback linearizable and exhibit nonholonomic constraints. Therefore new design methodologies should be explored.

Thus, nowadays such control problems of underactuated marine vehicles motivate the development of new nonlinear control design methodologies. Control designs of tracking, point stabilization, path following for some kind of marine vehicles, or dynamic positioning (DP) for offshore systems are examples of these types of problems.

Dynamic positioning (DP) is required in many offshore oil field operations, such as drilling, pipe-laying, tanking between ships, and diving support. Critical for the success of a dynamically positioned ship is its capability for accurate and reliable control, subject to environmental disturbances as well as to configuration related changes, such as a reduction in the number of available actuators. Furthermore, robustness criteria must be considered (Muñoz et al. 2006).

The problems of motion control for marine vehicles addressed in the literature (Aguiar and Hespanha 2003) can be roughly classified in three groups:

- Point stabilization, where the goal is to stabilize a vehicle at a given point, with a desired orientation.
- Trajectory tracking deals with the case where a vehicle must track a time-parameterized reference.
- Path following refers to the problem of making a vehicle converge to and follow a given path, without any temporal specifications.

The degree of difficulty involved in solving these problems is highly dependent on the configuration of the vehicle.

Point stabilization presents a true challenge to control system designers when the vehicle has nonholonomic constrains since, as pointed out in Brocket et al. (1983), there is no smooth (or even continuous) constant state-feedback control law that will do the job

For fully actuated systems, the trajectory tracking problem is now reasonably well understood and satisfactory solutions can be found in the standard non linear control textbooks.

For underactuated vehicles trajectory tracking is still an active research topic (Aguiar and Hespanha, 2003). Actuated systems are usually costly and often not even practical (due to weight, reliability, complexity, and efficiency considerations) and for this reason the interest in the study of this kind of problems. An interesting benchmark problem is described in (Aranda et al. 2006a), where control specifications for a hovercraft model are given, and a possible solution with a simulation environment for testing is presented in (Aranda et al. 2006b).

Path following control has received relatively less attention than the other two problems. See for example Samson (1995) and references. Path following systems for marine vehicles have been reported in Encarnação and Pascoal (2001) with a path following mode. Here the vehicle forward speed does not need to be controlled accurately, since just orientating the vehicle drives it to the path. Typically, smoother convergence to a path is achieved in this case when compared with the performance obtained with trajectory tracking controllers, and the control signals are less likely to be pushed to saturation.

Other kinds of control problems is concerning to the attenuation of non desirable movement in fast ships, attenuation of the vertical and lateral motions. The study of this problem is in relation with one of the most unpleasant aspects of sea transport that is the motion sickness suffered by both passengers and crew. This is a result of the accelerations associated with the induced roll, heave and pitch motions. (de la Cruz, et al. 2004). A solution to vertical stabilization of a fast ferry based in a multivariable QFT design is given in Aranda et al. (2005a), the extension to couple of lateral and vertical dynamic is done in Aranda et al (2005b). A methodology to identification of multivariable models based in genetic algorithm and non linear optimization procedure, with application to a high speed craft, is presented in Aranda et al (2004a). Extension and revision of this methodology is done in Aranda et al (2005c). A control oriented model for a high speed craft is obtained in Esteban et al. (2004).

In this chapter, we show some problems concerning to tracking and stabilization. In section 2, the benchmark problem for a RC-hovercraft is described. In section 3, robust control for a moored platform model is studied, and in section 4 the attenuation of non desirable movements are considered for a fast ferry.

2 Control of an Underactuated Vehicle

The task of designing controllers for underactuated marine vehicles is very challenging and has received increasing attention in the past few years. These vehicles exhibit complex hydrodynamic effects that must be taken into account during the control design. It should be highlighted that many marine vehicle models exhibit a drift vector field that is not in the span of the input vector fields and because of this, input transformations are not used to bring them to driftless form.

The past few decades have witnessed an increased research effort in the area of trajectory tracking control for underactuated autonomous vehicles. Trajectory tracking problems are concerned with the design of control laws that force a vehicle to reach and follow a time parameterized reference (i.e., a geometric path with an associated timing law).

Hovercrafts are a type of vehicle with a structure model similar to marine vehicles. A model for a nonlinear underactuated hovercraft was obtained from the ship model in (Fossen 1994). In this model, the hovercraft is equipped with two propellers that provide the thrust to move the vehicle forward (and backward) and to make it turn. The main difference with respect to a two-wheel mobile robot is that a hovercraft can move freely sideways even though this degree of freedom is not actuated.

Numerous control algorithms for controlling underactuated vessels have been examined and analysed in the specialized literature. For example, Fantoni et al. (2000) presents two control laws. The first one controls the velocity of the hovercraft. The other one proposes strategies for positioning the hovercraft at the origin.

Pettersen and Egelan (1996) developed a stability result involving continuous time-varying feedback laws that exponentially stabilize both the position and orientation of a surface vessel having only two control inputs.

In Fossen et al. (1998) it is considered a nonlinear ship model including the hydrodynamics effects due to time-varying speed. A backstepping technique for tracking control design is employed.

Bullo and Leonardo (1998) develop high-level motion procedures which solve point-to-point reconfiguration, local exponential stabilization and static interpolation problems for underactuated vehicles.

Strand et.at. (1998) propose a stabilizing controller by a locally asymptotically convergent algorithm based on $H\infty$ -optimal control.

Berge et al. (1998) develop a tracking controller for the underactuated ship using practical feedback linearization. The control law makes the position and velocities converge exponentially to the reference trajectory, while the course is not controlled.

Aguiar and Hespanha (2003) develop a nonlinear Lyapunov-based tracking controller and prove to exponentially stabilize the position tracking error to a neighbourhood of the origin that can be made arbitrarily small.

The problem of point stabilization of marine vehicles like hovercrafts that exhibit non-holonomic restrictions is so challenging, because as Brocket et al. (1983) showed, there is no smooth (or even continuous) constant state-feedback control law that stabilizes the system in a desired point in the state space.

The main problem for stabilization of underactuated hovercrafts is that any linearization of the system around an equilibrium point generates an uncontrollable system. This is due to the fact that there are no forces that allow controlling the drift velocity. This problem is related in Fantoni et al. (2000) that shows that the linear system is only controllable for a non-zero angular velocity. They also propose a controller that use yaw angle velocity as a virtual input to obtain a discontinuous control law for stabilization.

Another approach to the stabilization problem uses smooth time-varying control laws. An example of this technique is showed in Pettersen and Nijmeijes (2001) that uses a continuous periodic time varying feedback law.

Another author had applied the latest control techniques to the problem of point stabilization. An example of these techniques is showed in Tanaka et al. (2001) where a switching fuzzy control is implemented achieving stabilization in an RC hovercraft. Another example is in Segudii and Ohtsuka (2002) where stabilization is achieved using a model controller predictive with receding horizon strategy. The greater advantage of the last technique is that allows including saturation restrictions in control signals during the control design.

In our laboratory, control algorithm for tracking and point stabilization are analysed using a radio control hovercraft. Previous to the laboratory experimental validation with the radio controlled vehicle, simulations are carried out using Matlab and Easy Java Simulations (Aranda et al. 2006a) with the goal of comparing different control strategies and test diverse conditions. As a first approach, benchmark problems have been developed (Aranda et al. 2006b), a non linear control for both, tracking and point stabilization problems, was designed; also an assessment is designed as verification of the robust and performance criteria. Now, in our laboratory, techniques based in a non-linear multivariable QFT methodology are studied for this problem.

In the next subsections, the model of the hovercraft and the benchmark problem are described.

2.1 Hovercraft Model

The model system is a radio control hovercraft equipped with two longitudinal thrusters to control speed and turning as shown in Figure 1. The impulse of both motors is asymmetric and is greater when the hovercraft moves forwards than backwards.

Fig. 1. R.C. Hovercraft.

Figure 2 shows a schematic model: X and Y are the fixed inertial reference system axes, X_B and Y_B the body reference system axes, $\vec{u}$ *and* $\vec{v}$ the surge and sway velocities, θ is the orientation angle and Ψ the drift angle (Aranda et al. 2006b).

The hovercraft has three degrees of freedom, two associated with the movement in the plane of its centre of masses (x, y), and one more associated to its orientation θ. u_1 and u_2 variables are the forces of the thrusters and r is the distance between the centre of the fan and the symmetry axis that cuts to the centre of mass (x,y).

The vehicle is underactuated because it has more degrees of freedom than control actions. This means that is not possible to control the surge velocity because of the impellers configuration.

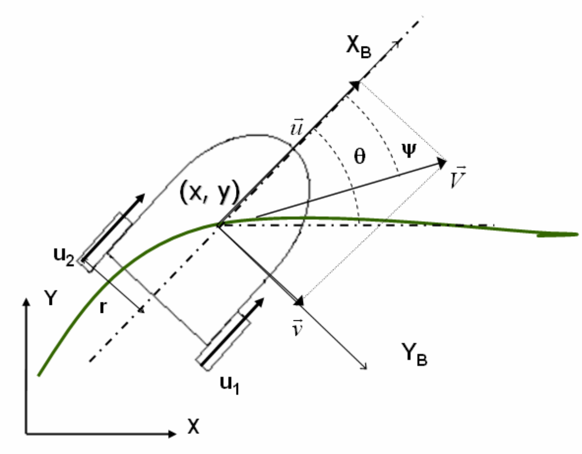


Fig. 2. Schematic model.

The dynamic equations are obtained in the fixed inertial reference system by direct application of Newton's laws.

$$m\frac{d^2x}{dt^2} = F_x = (u_1 + u_2)\cos(\theta) - \mu_T v_x \tag{1}$$

$$m\frac{d^2y}{dt^2} = F_y = (u_1 + u_2)\sin(\theta) - \mu_T v_y \tag{2}$$

$$J\frac{d^2\theta}{dt^2} = \tau = r(u_1 - u_2) - \mu_R w \tag{3}$$

Where m is the mass of the vehicle, J the moment of inertia, μ_T and μ_R the coefficients of viscous and rotational friction respectively. The system can be defined by the state vector [x, y, v_x, v_y, θ, w].

The parameters have been experimentally obtained by measurements in the real system. The mass, the force of thrusters and r have been measured directly. The friction coefficients are obtained from measurements of the maximum linear and rotational velocity. The nominal values and their uncertainties are shown in Table 1.

Table 1. Experimental parameters and uncertainties

Parameter	Value and uncertainties
m	0.894 ± 0.001 Kg
J	0.0125 ± 0.0050 Kgm2
r	0.050 ± 0.001 m
u_{max}	0.615 ± 0.008 N
u_{min}	-0.300 ± 0.008 N
μ_T	0.10 ± 0.01 Kg/s
μ_R	0.050 ± 0.001 Kgm2s

2.2 Benchmark Problems

A benchmark problem has been defined for each of the three points indicated in the introduction (point stabilization, tracking and path following) (Aranda et al. 2006a). The evaluation criteria for each problem are organized in three groups.

2.2.1 Performance Criteria

Point stabilization: The specifications for a displacement of a distance D≤1m in the set point without a change in direction are an overshoot M_p<5% and a setting time t_s≤20s, whereas for D>1m M_p<20%.

For a change of reference angle Δθ=90º without displacement the controller must verify an overshoot M_p<10% and a setting time t_s≤5s.

Trajectory tracking: a specification is set for the steady-state tracking error $TE_{ss} = lím_{t\to\infty} \left\| \vec{x}(t) - \vec{x}_r(t) \right\|$ for each trajectories defined in Table 2.

Path following: For a desired speed and a predefined trajectory, the lateral deviation must be reduced to 5% in a setting time t_s<20s. In addition the drift angle Ψ must be reduced.

Table 2. tracking specifications.

Trajectory	Specification
Circle, Radius=2m, V=0.2m/s	TE_{ss}<10cm
Circle, Radius=2m, V=0.8m/s	TE_{ss}<30cm
Straight line, V=0.2m/s	TE_{ss}<5cm
Straight line, V=0.8m/s	TE_{ss}<10cm

2.2.2 Robustness Criteria

Stability and performance must be satisfied by the controller for all the model parameter range shown in Table 1.

2.2.3 Actuator Effort Criteria

For the trajectories defined in section 3,1 the control signals must not exceed saturation limits of the actuators $u \in [u_{\min}, u_{\max}]$.

2.3 Example Controller

The controller inputs are the system state and the references vector [x_r, y_r, v_{xr}, v_{yr}, θ_r, w_r, d, l]. The controller outputs must be the control actions u_1 and u_2.

The example controller that is implemented by defect in the model consists of two cascade controllers PD (see Figure 3). The L1 controller takes the position errors and calculates the forces that must be applied in the fixed coordinates system to follow the trajectory. These forces are the references for the L2 controller that calculates the angle of the force with the x-axis and control the orientation to follow this angle. The outputs of L2 are the control signals u_1 and u_2 that must be applied to obtain to the force and the suitable direction θ. L2 obtains the direction from the state to control the hovercraft.

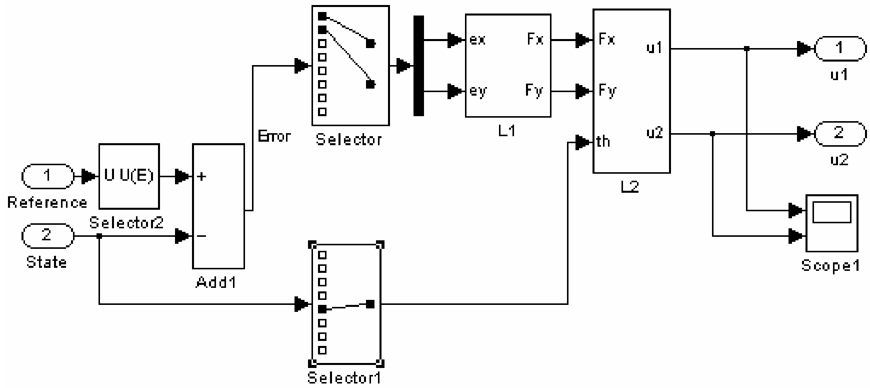


Fig. 3. Example of controller.

The parameters of the controllers are optimized for a predetermined trajectory (circle with a radius of two metres with 0.2m/s of velocity). The design and behaviour of the example controller is shown in (Aranda et al. 2006b).

3 Dynamic Positioning of a Moored Platform

Moored floating platforms are used for drilling and exploration activates and they require high degree of precise positioning to perform optimally with associated facilities. They are subjected to combined environmental loads of waves, wind and current while in service, which affects their stability in addition to positioning.

Dynamic analysis of a floating moored platform is therefore carried out to determine its response to environmental loads. This analysis could be done in the time domain or the frequency domain. A floating platform is associated with lots of non-linearities, which are linearised in the frequency domain simulations. In the time domain analysis however, the non-linearities are modelled with the intention of making more accurate analysis but this makes the computation complex and require a great deal of computer time.

In our group, a moored floating model is considered as a testing platform to analysis different robust synthesis methods for Dynamic Positioning.

The model is a single-input/multi-output (SIMO) linear time invariant (LTI) system with a single degree of freedom.

This system has been examined and analysed in the specialized literature, in which is possible to find several robust control methods. For example, (Scherer, et al., 1997) presents an overview of a linear matrix inequality (LMI) approach to the multiobjective synthesis of linear output-feedback controllers. A multiobjective H_2/H_∞ is proposed to specify the closed-loop objectives in terms of a common Lyapunov function.

In (Revilla, 2005), this system is used to validate the results obtained in the study about synthesis of reduced-order controllers based on LMI optimization. In Nakamura, et al. (2001) the problem was formulated in the framework of a multimodel-based design of the $H\infty$ control law with pole region constraint. Methodology based on LMI was used to solve the problem. And in (Muñoz-Mansilla et al. 2006), a multivariable robust QFT controller is used to stabilize the moored platform.

The next subsections show the control problem and a solution by QFT technique.

3.1 Control Problem

The system consists of a floating platform that is anchored to the bottom of the ocean and equipped with two thrusters, as it is showed in Figure 4 (the model of a replica of this system and previous control is described in Kajiwara et al. 1995). The objective is achieving an appropriate thrusters control in order to minimize the drift Y resulting from the wave action.

The model of the system has two outputs $\boldsymbol{y}$ (the horizontal drift Y and angular deviation from the vertical axis ϕ), one control input u (the force delivered by the thrusters F_u) and two disturbance inputs $\boldsymbol{d}$ (the force F and the torque M from the wave action). Therefore a single degree of freedom (DOF) SIMO system is presented, with one single input F_u and two outputs (Y, ϕ).

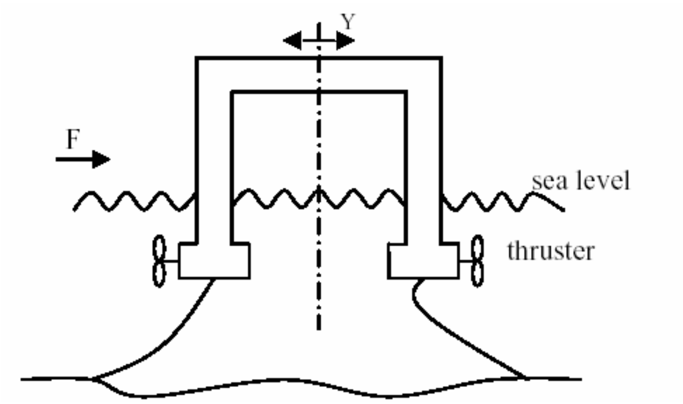


Fig.4. Moored floating platform

For design purposes, the system transfer function can be described as:

$$\boldsymbol{y} = \mathbf{P_{plant}(s)} \cdot u + \mathbf{P_d(s)} \cdot \boldsymbol{d} \tag{4}$$

$$u = -\,\mathbf{G_{control}(s)}\,\boldsymbol{y} \tag{5}$$

Where $\mathbf{P_{plant}(s)}$ is a transfer functions matrix (2x1) that connects the input *u* with the output ***y***, and $\mathbf{P_d(s)}$ is a transfer functions matrix (2x2) that connects the disturbance ***d*** with the output ***y***. The control structure is schematically displayed in Figure 5.

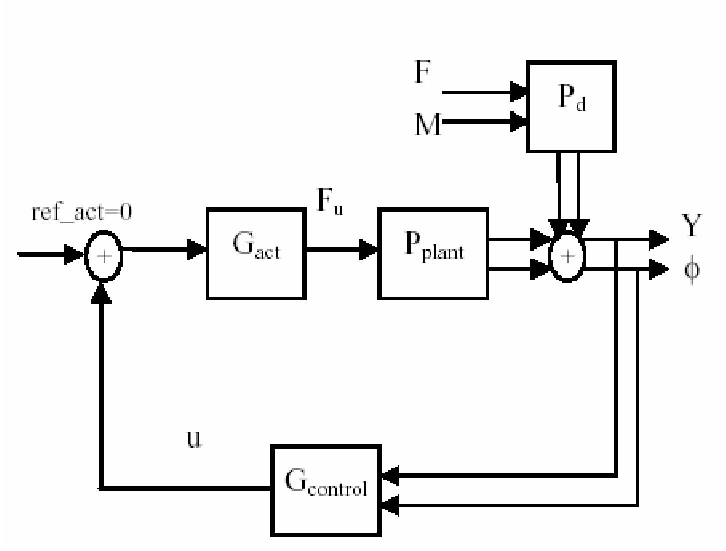


Fig.5. Single DOF SIMO system with disturbances at the plant's output

In these conditions, the problem of interest is how to design the controller $G_{control}$. The control objectives are:

- Reduce the drifting action F_2 by using the actuators control.
- Maintain the horizontal drift $|Y| < 0.025$m
- Maintain the angular deviation $|\phi| < 3$ degrees
- Keep $|F_u| < 0.25$ N
- Make sure that the thrusters have no response to the high-frequency component F_1.

An interesting question is added to the position control design because the plant has less degree of freedom for actuation, it is an underactuated system, and is more difficult to control.

3.2 A Multivariable QFT Controller for the Moored Platform

The foundation of QFT is the fact that feedback is principally needed when the plant is uncertain and/or there are disturbances acting on the plant.

Taking into account all this; the challenge is to study the effectiveness of the QFT technique to accomplish the dynamic positioning system (Muñoz et al 2006).

The QFT design procedure involves three basic steps: *i*) computation of QFT bounds, *ii*) design of the controller (loop shaping), and *iii*) analysis of the design.

QFT converts close-loop magnitude specifications into magnitude constraints on a nominal open-loop function (QFT bounds). A nominal open-loop function is then designed to simultaneously satisfy its constraints as well as to achieve nominal closed-loop stability (loop shaping). It is defined the open-loop function $L(j\omega)$ as the product of the controller transfer function and the plant transfer function.

In any QFT design, it is necessary to select a frequency array for computing bounds. In the case of the platform plant, the range of frequencies that belongs to the seaway spectrum will be $\omega \in [0.1, 10]$.

The specifications must be given in terms of frequency response. For the particular case of the design of the dynamic positioning system for the moored platform model, the specifications ($|Y| < 0.025$m, $|\phi| < 3$ degrees) are given in temporal domain. Therefore, it is necessary to translate these constraints into frequency domain specifications. The QFT specifications used are: the gain and phase margins stability, the output disturbance rejection and the control effort.

The control law of the system in Figure 5 is:

$$u = -G_{control} \cdot \begin{pmatrix} Y \\ \phi \end{pmatrix} = -\begin{pmatrix} k1 & k2 \end{pmatrix} \cdot \begin{pmatrix} Y \\ \phi \end{pmatrix} \tag{6}$$

Solving (4), it yields one equation with two unknown quantities, *k1* and *k2*.

$$\begin{aligned} &(\pi_{13} + k1) \cdot Y + (\pi_{23} + k2) \cdot \phi = \\ &= (\pi_{13} p11 + \pi_{23} p21) \cdot F + (\pi_{13} p12 + \pi_{23} p22) \cdot M \end{aligned} \tag{7}$$

The control design process is based on this equation, which aids in transforming the problem into the design of two sequential SISO systems. Thus, it is solved by an iterative multi-stage sequential procedure, in such a way that the solution of *k1* in the first system is used in the design of *k2* in the second system, and vice versa. The stages repeat successively up to *k1* and *k2* meet the objectives for the SIMO system. Finally, the control design procedure has been completed in five stages.

Temporal responses of the SIMO system (Figure 5) in closed loop dynamic are shown. Figures 6 and 7 compare both the outputs Y and ϕ with and without control respectively. It is shown that the control achieves the positioning system.

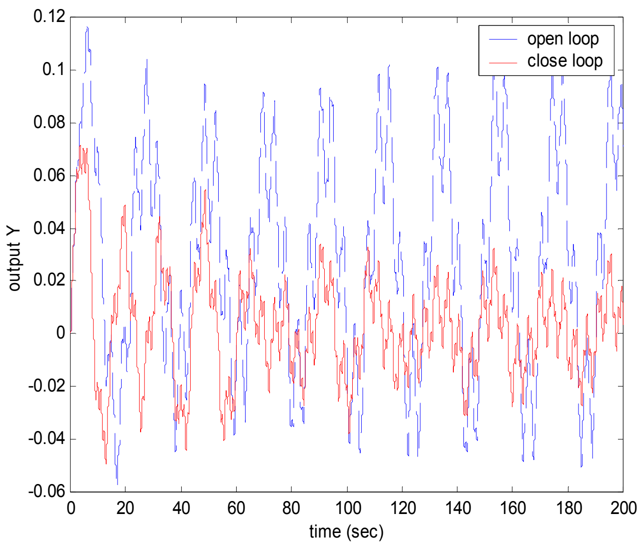


Fig. 6. Comparison of temporal responses *Y (a)* with and without control

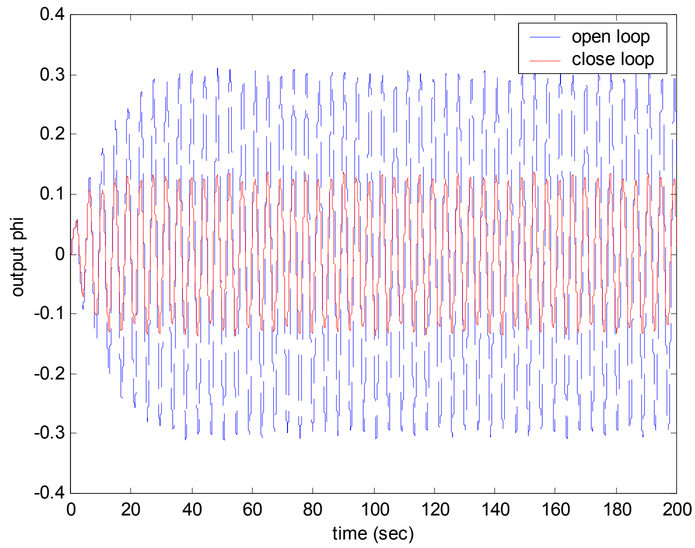


Fig. 7 Comparison of temporal responses ϕ *(b)* with and without control

4 Stabilization of Vertical and Lateral Movements in a Fast Ferry

The interest on fast ships for cargo and passenger transportation was growing during the past decade. Different designs have been considered, and a significant attention has been focused on fast monohull displacement ships (Allison, et al., 2004).

The next history is an interesting introduction to main problems in the building of fast ferries in relation with the seasickness (Laertius): Anacharsis, brother of Caduides the king of the Scythians, was a philosopher who traveled around the East Mediterranean and the Black Sea in the 6th century BC. His mother was Grecian woman and the contemporaneous Greeks said him that he exhorted moderation and good criteria in everything he did, for example saying “the vine bears three clusters of grapes: the first wine, pleasure; the second, drunkenness, the third, disgust”. As a seaman, he had to travel in different kinds of sea conditions and he had one of the best references, with the same philosophy, listened about seasickness “people may be divided into three classes; the living, the dead and the seasick”.

Therefore, the main objectives in the design and built of high speed crafts are the passenger comfort and the vehicle safety. The vertical accelerations associated with roll, pitch and heave motions are the principal cause of motion sickness.

The ship considered in our researches is a fast ferry with the following characteristics (de la Cruz et al. 2004): 110m. length, 1250 passengers, deep-V monohull, aluminium made, able to get 40 knots or more. Fig. 8 shows a photograph of the ship.

Fig. 8. Fast ferry

Previous researches of the work group have studied the longitudinal and transversal dynamics separately and next the coupled of both dynamics. Firstly, it has been studied heaving and pitching motion for the case of head seas (μ=180deg) (Aranda, et al., 2004a), modeled actuators and designed different controllers was done, (Aranda, et al., 2002a, Aranda, et al., 2002b, Aranda et al. 2005a, Diaz et al. 2005, Esteban et al 2001, Esteban et al. 2005), in order to achieve heave and pitch damping and with successful results. And secondly, it has been analyzed the rolling response for the case of lateral waves (μ=90deg) (Aranda, et al., 2004b) and in the same way, it has been carried out the actuators modeling and controller designing (Aranda et al. 2005b, Aranda et al. 2004b) .

4.1 The Control Problem

The goal is to reduce longitudinal and transversal motions of the fast ferry system with three coupled degrees of freedom.

Three modes of the system are analyzed: the heave and pitch motions (vertical dynamics) and the roll motion (horizontal dynamics).

The actuators employed for the vertical dynamic control consisted of active stabilized surfaces, one T-Foil on the bow and two flaps on the stern. The control surfaces employed for the roll control were two fins.

The model of the system has two outputs: the roll angular velocity ω_{roll} and the vertical acceleration a_{cv}. The input disturbances are the seaway. The control inputs are the angles of attack α_P and α_H of flaps and T-Foil, and α_R for the lateral fins. The coupling of the modes is considered as a consequence of the control surface action in different incidence angles of the seaway.

The mathematical models of the three modes of the craft and actuators are obtained from system identifications methods (Aranda 2004a, Aranda et al. 2005c).

Therefore, the problem of interest is how to design the controllers for these actives surfaces to achieve a reduction of the Motion Sickness Incidence (MSI), given by equation (8) (de la Cruz, et al. 2004):

$$MSI = 100 \cdot \left[0.5 \pm erf\left(\frac{\pm \log_{10}(|\overline{\ddot{s}_3}|/g) \mp \mu_{MSI}}{0.4} \right) \right] \tag{8}$$

This aim can be translate to different specifications, in accordance whit the design methodology chosen. For example (Rocio CAMS), we can consider the following specifications:

- system stability,
- heave, pitch, roll reduction,
- no saturation on T-Foil, flaps and fins.

4.2 An Example Control

A control system must perform mainly three functions. The firts is to assure stability, the second is to attenuate seaway-induced motions, and the third is to assure the safety of the ship and its passengers. Different solutions were given, (see for example the references Esteban et al. 2004, Aranda et al 2002a, 2004b, 2005a, 2005b, Díaz et al. 2005).

As an example, we can considered the design of a PI scheduling for each speed of the craft, in the particular case of roll reduction.

For the particular case of V = 40 knots, figure 9 shows the root locus of the system with the controller PI $G_{PI40}(s)$. The transfer function of this controller is given by expression (9).

$$G_{PI40} = 2.2\frac{50s+1}{50s} \tag{9}$$

Table 1 shows the values of J and roll reduction percentage for the cases of regular wave with frequency 1 rad/s and irregular wave with SSN = 5 with this controller.

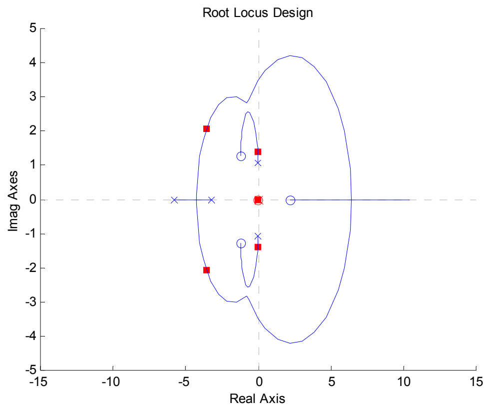


Fig. 9. Root locus of the system with G_{PI40}

Table 1. Value of J and reduction for control PI. V=40 knots

wave type	J	roll reduction (%)
regular ω = 1rad/s	0.29	82.69
irregular SSN=5	0.82	69.01

5 Conclusions

The problems of tracking and stabilization of underactuated vehicles, dynamic positioning, and control for stabilization in high speed craft are example of the problems solved in our group. For each of these problems, a description of the problem and an example of one solution is showed. There are different solutions and a complete description of each case in the references of our group.

Acknowledgements

This development was supported by CICYT of Spain under contract DPI2003-09745-C04-01.

References

Aguiar A. P. Hespanha J. P. 2003. Position tracking of underactuated vehicles. In Proc. of ACC'03 - American Control Conference, Denver, CO, June 2003.

Allison, J.L., B.G.Forstell, D.R.Lavis, J.Purnell, CDI Marine. 2004. The Influence of New Technology on the Design and Manufacture of High Speed Craft. RINA International Conference on High Speed Craft, 17-18 November.

Aranda J., de la Cruz J. M., Díaz J.M., Dormido Canto S. 2002a. Qft Versus Classical Gain Scheduling: Study For A Fast Ferry. 15th IFAC World Congress b'02.

Aranda J., Revilla J., Diaz J.M. and Ruipérez P. 2002b. Experimental and robustness analysis of a multivariable control for vertical dynamic of a high speed craft. In Proceedings of the 2002 IEEE International Conference on Control Applications September 18-20, 2002 • Glasgow, Scotland, U.K. Pp.: 1070-1075.

Aranda J., de la Cruz J.M., Díaz J.M. 2004a. Identification of Multivariable Models of Fast Ferries. European Journal of Control. Vol.10, n. 2. May 2004. Pp: 187-198.

Aranda J., Muñoz-Mansilla R., Díaz J.M. 2004b The problem of the vertical movement control with roll movement in fast feries. In Proc. of CAMS 2004 IFAC Control Applications in Marine Systems. Ancona July 7-9. Pp. 339-344.

Aranda J., de la Cruz J.M., Díaz J.M. 2005a.Design of a multivariable robust controller to decrease the motion sickness incidence in fast ferries. Control Engineering Practice. Vol 13/8.. 2005. Pp: 985-999.

Aranda J., Muñoz Mansilla R., Díaz J.M., Dormido Canto, S. 2005b. Robust Control for the Coupling of Lateral and Longitudinal Dynamics in High-Speed Crafts. In Proc. of IFAC World Congress. 2005.

Aranda J., Muñoz R., Dormido Canto S., Díaz J.M., Dormido, S. 2005c. An Análisis of Models Identification Methods for High Speed Craft. Journal of Maritime Research. Vol. II No. 1. 2005. Pp. 51-67.

Aranda J., Chaos D., Dormido-Canto S., Muñoz R., Díaz J.M.. 2006a. Benchmark Control Problems for a Non-Linear Underactuated Hovercraft: A Simulation Laboratory for Control Testing. In Proc. of Advance Control Education ACE 2006. Madrid

Aranda J., Chaos D., Dormido-Canto S., R.Muñoz, Díaz J.M. 2006b. A Control for tracking and stabilization o fan underactuated non-linear RC hovercraft. In Manoeuvring and Control of Marine Craft MCMC 2006. Lisbon.

Berge S. P., Ohtsu K., and Fossen T. I.. 1998. Nonlinear tracking control of underactuated ships minimizing the cross-track error". In Proc. of the IFAC Conf. on Control Applications in Marine Systems (CAMS'98), Fukuoka, Japan, October 1998. pp 141-147.

Brockett R.W., Millman and Sussman H. J. 1983. "Asymptotic stability and feedback stabilization". Diferencial Geometric Control Theory. Eds, Boston, pp 181-191.

Bullo and Leonanrd N.E. 1998.. Motion primitives for stabilization and control of F. underactuated vehicles. Preprints of the 4th IFAC NOLCOS'98, Enschede, The Netherlands, July 1998, 1:133-138.

De la Cruz J.M., Aranda J., Giron-Sierra J.M., Velasco F., Esteban S., Diaz J.M., de Andres-Toro B. 2004. Improving the Comfort of a Fast Ferry, Smoothing a ship's vertical motion with the control of flaps and T-foil. IEEE Control System Magazine. Abril 2004. Pp: 47-60.

Díaz J.M., Dormido S., Aranda J. 2005. Interactive computer-aided control design using Quantitative Feedback Theory: The problem of vertical movement stabilization on high-speed ferry. International Journal of Control. Vol. 78, issue 11. 2005. Pp.: 793-805

Encarnação P., and Pascorar A. 2001. Combined trajectory tracking and path following for marine craft. In Proceedings of the nineth Mediterranean conferences on control and automation. Dubrovnik, Croatia.

Esteban. S., Girón Sierra J.M., de la Cruz J. M., de Andrés B. 2001. Predictive perturbation cancelling for seakeeping improvement of a fast ferry. In Proc. of European Control Conference. ECC'01. Porto, Portugal. Septiembre 2001.

Esteban, S.; Giron-Sierra, J. M; De Andres-Toro, B.; De La Cruz, J. M. 2004. Development of a Control-Oriented Model of the Vertical Motions of a Fast Ferry. Journal of Ship Research. Vol. 48, no. 3, pp. 218-230. Sept. 2004.

Esteban S., Girón-Sierra J.M., Andrés-Toro B., Besada E., de la Cruz J.M.. 2005. Fast Ship Model for Seakeeping Improvement Studies Using Flaps and T-Foil. Mathematical and Computer Modelling. Edited Elservier.

Fantoni I., Lozano R., Mazenc F., and Pettersen K.Y. 2000. Stabilization of a nonlinear underactuated hovercraft. Int J. of Robust and Nonlinear Control, 10:645-654.

Fossen T. I.. 1994. Guidance and Control of Ocean Vehicles. Chichester: John Wiley & Sons Ltd.

Fossen T. I., Godhavn J.M., Berge S. P., and Lindegaard K.P..1998. Nonlinear control of underactuated ships with forward speed compensation. Preprints of the 4th IFAC NOLCOS'98, Enschede, The Netherlands, July 1998, 1:121-126.

Kajiwara, H., W. Koterayama,, M. Nakamura, P. Gahinet. 1995. LMI-based design of dynamic positioning control laws for a moored floating platform model. Proc. 14th SICE Kyushu Branch Annu. Conf. pp:181-184.

Laertius, D. Live of the Philosophers, classical

Muñoz-Mansilla R., Aranda J., Díaz J.M., Dormido-Canto S. and Chãos D. 2006. Robust control design by QFT methodology for a dynamic positioning problem of a moored floating platform . In Proc. of 7th IFAC Conf. on Manoeuvring and Control of Marine Crafts (MCM'06), Lisboa, Portugal, September 2006.

Nakamura, M., W. Koterayama, H. Kajiwara, T., Hyakudome.. 2001. Model Experiments on Dynamic Positioning System Using Gain Scheduled Controller. International Journal of Offshore and Polar Engineering. Vol 11, no 4.

Pettersen K.Y. and Egelan O. 1996. Exponential stabilization of an underactuated surface vessel. Proc. Of the 35th IEEE Conference on Decision and Control, 1996, pp 967-971.

Pettersen K. Y. and Nijmeijer H. 2001. Underactuated ship tracking control: theory and experiments, Int. J. Control, Vol. 74, Nº. 14, 1435-1446.

Revilla J. 2005. Contribución a la Síntesis de Controladores Robustos mediante técnicas LMIs. PhD thesis. UNED. Madrid.

Samson C. 1995. Control of Chained Systems Application to Path Following and Time-Varying Point-Stabilization of Mobile Robots. IEEE Transactions on Automatic Control, VOL. 40, NO. 1, January 1995.

Scherer. C., P. Gahinet, M., Chilali.. 1997. Multiobjetivo Output-Feedback Control Via LMI Optimization. IEEE Transactions on Automatic Control, Vol 42, pp :896:911.

Seguchi H. and Ohtsuka T. 2002 Nonlinear Receding Horizon Control of an RC Hovercraft. Proceedings of the 2002 IEEE International Conference on Control Applications. September 18-20, 2002 , Glasgow, Scotland, U.K.

Strand J.P., Ezal K., Fossen T.I, and Kokotovic P.V. 1998. Nonlinear control of ships: a locally optimal design. Preprints of the 4th IFAC NOLCOS'98, Enschede, The Netherlands, July 1998.

Tanaka K., Iwasaki M., and Wang H. O.2001.Switching Control of an R/C Hovercraft: Stabilization and Smooth Switching. IEEE Transactions on systems, man, and Cybernetics-Part B: Cybernetics, VOL. 31, NO. 6, December 2001.

CHAPTER 2

Interactive software tools for robust control: application to marine systems

J. M. DIAZ[1], S. DORMIDO[1], J. ARANDA[1], S. DORMIDO CANTO[1], R. MUÑOZ[1] and D. CHAOS[1]

[1]Departamento de Informática y Automática, UNED, Madrid
josema@dia.uned.es

This work presents the main features of QFTIT and TIG two interactive software tool for robust control design using the QFT methodology. The main advantages of QFTIT and TIG compared to other existing tools are its ease of use and its interactive nature. These tools are freely available in the form of an executable file for Windows and Mac based platforms. In order to illustrate its use, this paper also includes an example; a robust controller is designed to stabilize the vertical movement of a high-speed ferry.

1 Introduction

Interactive tools are considered a great stimulus for developing the control engineering intuition. At present, a new generation of interactive software control packages have created an interesting alternative in comparison with the traditional approach. In this sense innovative and interesting ideas and concepts were implemented by Prof. Åström and coll. at Lund such as the concept of dynamic pictures and virtual interactive systems (Wittenmark et al., 1998). The main objective of these tools is to involve the users in a more active way in the analysis and design processes.

In essence, a dynamic picture is a collection of graphical windows that are manipulated simply by using the mouse. Users do not have to learn or write any sentences. If we change any active element in the graphical windows an immediate recalculation and presentation automatically begins. In this way we perceive how these modifications affect the result obtained.

These kinds of tools are based on objects that allow direct graphic manipulation. During these manipulations, the objects are immediately updated. Therefore the relationship among the objects is continuously maintained. Ictools and CCSdemo (Johansson et al., 1998) (Wittenmark et al., 1998) developed at the Department of Automatic Control at Lund Institute of Technology, and SysQuake at the Lausanne Federal Polytechnic School Automatics Institute (Piguet et al., 1999), are good examples of this new philosophy to produce a new family of interactive Computer-Aided Design (CAD) packages in automatic control.

Very often our work as control engineers is reduced to tune some design parameters using a trial and error procedure following an iterative process. Specifications of the problem are not normally used to calculate the value of the system parameters because there is not an explicit formula that connects them directly. This is the reason for dividing, each iteration, into two phases. The first one, often called synthesis, consists in calculating the unknown parameters of the system taking a group of design variables (which are related to the specifications) as a basis. During the second phase, called analysis, the performance of the system is evaluated and compared to the specifications. If they do not agree, the design variables are modified and a new iteration is performed (Dormido, 2004).

It is possible, however, to merge both phases into one and the resulting modification in the parameters produces an immediate effect. In this way, the design procedure becomes really dynamic and the users perceive the degree of change in the performance criteria given for the elements that they are manipulating. This interactive capacity allows us to identify much more easily the objectives that can be achieved (Dormido, 2003).

Interactive design with instantaneous performance display goes one step further. In many cases, it is not only possible to calculate the position of a graphic element (be it a curve, a pole, a template or a bound) from the model, controller or specifications, but also to calculate a new controller from the position of the element. For instance, a closed loop pole can be computed by calculating the roots of the characteristic polynomial, which itself is based on the plant and controller; and the controller parameters can be synthesised from the set of closed loop poles if some conditions on the degrees are fulfilled.

This two-way interaction between the graphic representation and the controller allows the manipulation of the graphical objects with a mouse in a very natural form. Since a good design usually involves multiple objectives using different representations (time domain or frequency domain), it is possible to display several graphic windows that can be updated simultaneously during the manipulation of the active elements.

The philosophy of interactive design with instantaneous performance display offers two main advantages when compared with the traditional procedure (non-interactive approach). In the first place, it introduces from the beginning the control engineer to a tight feedback loop of iterative design. The designers can identify the bottlenecks of their designs in a very easy way and can attempt to fix them. In second place, and this is probably even more important, not only is the effect of the manipulation of a design parameter displayed, but its direction and amplitude also become apparent. The control engineer learns quickly which parameter to use and how to push the design in the direction of fulfilling better tradeoffs in the specifications. Fundamental limitations of the system and the type of controller are therefore revealed (Åström 1994, 2000) giving way to finding an acceptable compromise between all the performance criteria. Using this interactive approach we can learn to recognise when a process is easy or difficult to control.

In this paper, such features of new interactive control tools are exploited in the field of Quantitative Feedback Theory as a "user-friendly" way to learn the main concepts, solve any non-trivial robust control problem, and gain experience of how Quantitative Feedback Theory (QFT) technique works. The focus has been on explaining the main features of Quantitative Feedback Theory Interactive Tool (QFTIT) (*http://ctb.dia.uned.es/asig/qftit/*) and Template Interactive Generator (TIG) (*http://ctb.dia.uned.es/asig/tig/*) and its role to encourage and help users understand the QFT technique. The constant increase in computing speed and power certainly makes that initial idea a real prospect.

QFT is a very useful robust control design methodology created by I. Horowitz (Horowitz, 1963). QFT uses frequency-domain concepts to satisfy performance specifications and to handle plant uncertainty. The basis of QFT relies on the observation that feedback is needed mainly when the plant is uncertain and/or when there are uncertain input disturbances acting on the plant. At the moment, and to the best of our knowledge, no fully interactive QFT tools are presently available.

QFT has been successfully applied to solve different control problems in various fields of engineering (Houpis et al., 2006). In the field of marine systems, QFT has been used in fast ferries to decrease the motion sickness incidence (Aranda et al., 2002a, 2002b, 2005a) (Cruz et al, 2004) (Díaz, 2005), to attenuate the roll movement (Aranda et al., 2004), and to control the lateral and longitudinal dynamics (Aranda et al., 2005b). Besides, QFT has been used in moored floating platform for dynamic positioning (Muñoz et al., 2006), and in hovercraft for tracking and stabilization (Aranda 2006a, 2006b).

The paper is organised as follows. In section 2 a brief description of the basic concepts of Quantitative Feedback Theory is included. Section 3 describes the main features of QFTIT and TIG. Section 4 includes an example to illustrate the use of these tools. A robust controller is designed to stabilize the vertical movement of a high-speed ferry. Finally, section 5 offers some conclusions.

2 Basic Concepts of QFT Methodology

The design procedure using QFT has been described in a wide variety of articles and books (Horowitz 1963, 1992, 2001) (Houpis et al., 2006) (Yaniv, 1999). QFT is a methodology used in the design of control systems including uncertainties in the plant which is subject to external disturbance in the input and output of the plant as well as measurement noise. Figure 1 illustrates the basic idea behind QFT applied to a SISO system with a control structure with two degrees of freedom. *F* is the transfer function of a pre-filter acting on the reference input *r*. *C* is the controller which, depending on an error signal *e*, generates a control signal *u* over the set of transfer functions of the plant *P*. This set describes the uncertainty region of the plant's parameters. *P* may be subject to disturbances at its input *v* and/or at its output *d*. *H* is the measure sensor of the output signal *y*, which may be affected by a measurement noise *n*.

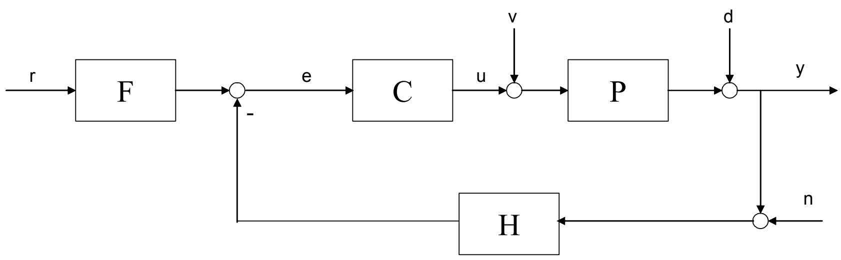


Fig. 1. Feedback System

The QFT method takes into consideration the quantitative information of the plant's uncertainty, robust operation requirements, robust tracking, expected disturbance amplitude and the associated damping requirement.

The controller *C* must be designed in such a way that the output variations *y*, which is a consequence of plant uncertainties *P*, is within specified tolerance boundaries. Furthermore, the effects of the disturbances *d* and *v* on the output must be acceptably small. On the other hand the pre-filter *F* is designed in order to perform the desired control of the reference signal *r*.

The design is performed using a Nichols diagram, defining a discreet set of trial frequencies Ω. This set is taken around the desired crossover frequency. As we are treating a family of plant instead of a single plant, the magnitude and phase of the plants in each frequency correspond to a set of points in the Nichols diagram. These sets of points form a connected region or a set of disconnected regions called "template". $\mathcal{T}(\omega_i)$ denotes a template computed at the frequency $\omega_i \in \Omega$. A large template implies a greater uncertainty for a given frequency. The templates and the working specifications are used to define the domain bounds within the frequency domain. The domain bounds set the limit of the frequency response of the open loop system.

Each specification contains bound definitions. Bounds are calculated using the corresponding templates and specifications. The different types of bounds are calculated in the following way:

- *Stability bounds*. By using templates and the specified phase margin.
- *Control bounds*. By using templates and the upper and lower limits of the response in the frequency domain.
- *Disturbance bounds*. By using the disturbance refusal specifications.
- *Effort control bounds*. By using templates and the specified control limits.

All the bounds computed at the same frequency $\omega_i \in \Omega$, associated at the different specifications are intersecting to generate a final bound $B(\omega_i)$ which includes the most restrictive regions of all the considered bounds.

The controller is designed by means of a loop-shaping process in the Nichols diagram. This diagram sketches the intersection of the bounds calculated for each of the trial frequencies and the characteristics of the open loop nominal transfer function $L_0(j\omega)=C(j\omega)\cdot P_0(j\omega)$.

The design is carried out by adding gains, poles and zeroes to the frequency response of the nominal plant, in order to change the shape of the open loop transfer function. By doing so, the boundaries $B(\omega_i)$ are kept for each $\omega_i \in \Omega$. The controller is the set of all the aforementioned items (gain, poles and zeroes).

Should there be any specification which corresponds to the control of the reference signal, a pre-filter F must be used. This pre-filter is designed in a similar way as the controller, the difference being in the use of the limits imposed in the frequency response control. In this case, the shaping may be carried out in the Bode diagram instead of the Nichols diagram.

The last step for the QFT design is the analysis and validation which includes not only the analysis in the frequency domain but also the simulations in the temporary domain of the resulting closed loop system.

The benefits of QFT may be summarised as follows:

- The outcome is a robust controller design that is insensitive to plant variation
- There is only one design for the full envelope and it is not necessary to verify plants inside templates
- Any design limitations are clear at the very beginning
- There is less development time in comparison to other robust design techniques
- QFT generalises classical frequency-domain loop shaping concepts to cope with simultaneous specifications and plants with uncertainties
- The amount of feedback is adapted to the amount of plant and disturbance uncertainty and to the performance specifications
- The design trade-offs in every frequency are transparent between stability and performance specifications. It is possible to determine what specifications are achievable during the early stages in the design process
- The redesign of the controller for changes in the specifications can be done very fast.

3 Basic Features of the Interactive Tools QFTIT and TIG

3.1 QFTIT

QFTIT is an interactive software tool which implements the QFT methodology. It highlights for its interactive nature and easy-of-use (Díaz et al., 2005b). In QFTIT, all that the designer has to do is to place the mouse pointer over the different items (see Fig. 2.) which the tool displays on the screen. Any action carried out on the screen is immediately reflected on all the graphs generated and displayed by the tool. In this way the users can quickly see the effects of their actions during the design. QFTIT has been built on Sysquake (Piguet, 1999) and it implements in an only very interactive GUI all the stages of the QFT methodology. Therefore, it is very adequate to novice designer that wants to learn QFT. But, it can be used for advanced designer to solve real robust control problem with QFT (Díaz et al., 2005a). It is important to remark that QFTIT is freely distributed as an executable file for Windows and Mac platforms. It does not need to install additional programs, like Matlab, to run QFTIT.

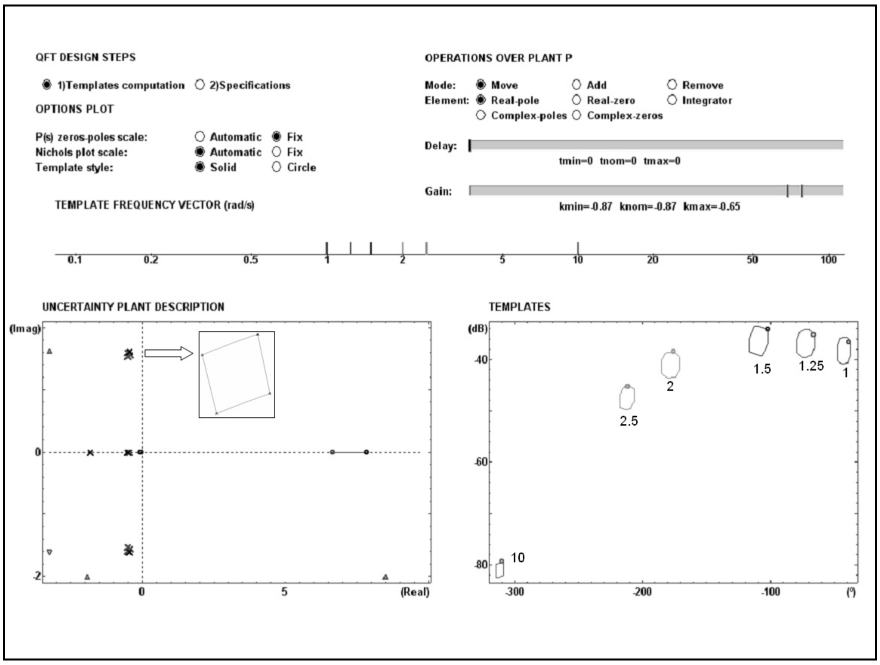


Fig. 2. Example of the QFTIT windows in the stage 1 (Templates computation)

The present version of QFTIT offers the user, amongst others, the chance of observing the changes instantly when some form of modification takes place in the available interactive objects:

- Variations produced in the templates when a change of the uncertainties of the different components of the plant or in the value of the template calculation frequency.
- Individual, grouped or intersected variation in the bounds as a result of the configuration of specifications, i.e., by adding zeroes and poles to the different specifications.
- The motion of the controller's zeroes and poles over the complex plane and the variation of its symbolic transfer function when the open loop transfer function is modified in the Nichols diagram.
- The change of shape of the open loop transfer function in the Nichols diagram and the variation of the expression of the controller's transfer function when any movement, addition or suppression of its zeroes or poles in the complex plane.

· The changes that take place in the temporary representation of the manipulated variable and in the controlled variable due to the variation of the nominal values of the different elements of the plant.
· The changes that take place in the temporary representation of the manipulated variable and of the controlled variable due to the introduction of a step perturbation in the input of the plant. The magnitude and the occurrence instant of the perturbation is configured by the user by means of the mouse.

Obviously, the present version of QFTIT has some limitations:

· It can only work with continuous SISO systems whose plant *P* is expressed in real factored form (RFF):

$$P(s) = \frac{K\cdot e^{-\tau s}\cdot\prod_{i=1}^{m}(s+z_i)\cdot\prod_{j=1}^{a}\left(s^2+2\cdot\delta_j\cdot\omega_{0j}+\omega_{0j}^2\right)}{s^N\cdot\prod_{l=1}^{n}(s+p_l)\cdot\prod_{q=1}^{b}\left(s^2+2\cdot\delta_q\cdot\omega_{0q}+\omega_{0q}^2\right)} \qquad (1)$$

where K, τ, z_i, δ_j, ω_{0j}, p_i, δ_q, ω_{0q} are independent variables which can take the following uncertainty in their value:

$$K \in [K_{\min}, K_{\max}] \subset \Re^- o \Re^+ \qquad (2)$$

$$\tau \in [\tau_{\min}, \tau_{\max}] \subset \Re^+ \qquad (3)$$

$$\tau \in [\tau_{\min}, \tau_{\max}] \subset \Re^+ \qquad (4)$$

$$z_i \in [z_{i\min}, z_{i\max}] \subset \Re \quad i = 1,\ldots,m \qquad (5)$$

$$p_l \in [p_{l\min}, z_{l\max}] \subset \Re \quad l = 1,\ldots,n \qquad (6)$$

$$\delta_j \in [\delta_{j\min}, \delta_{j\max}] \subset \Re^- o \Re^+ \quad j = 1,\ldots.,a \qquad (7)$$

$$\omega_{0j} \in [\omega_{0j\min}, \omega_{0j\max}] \subset \Re^+ \quad j = 1,\ldots,a \qquad (8)$$

$$\delta_q \in [\delta_{q\min}, \delta_{q\max}] \subset \Re^- o \Re^+ \quad q = 1,\ldots.,b \qquad (9)$$

$$\omega_q \in [\omega_{q\min}, \omega_{q\max}] \subset \Re^+ \quad q = 1,\ldots,b \qquad (10)$$

· According to this limitation, QFTIT only implements the algorithm by (Gutman et al., 1995) for the calculation of templates for RFF plants. The maximum number of templates which the tool can

support in the current version is 10. However, this number can be modified without any difficulty.
- It only woks with frequency-domain specifications.

QFTIT implements the QFT design methodology by considering a maximum of five design stages:

1) *Templates computation*. During this stage, the user defines the plant (1), by configuring the uncertainty of its components. Furthermore, the user also selects the set of trial frequencies Ω.
2) *Specifications*. In this stage, the user selects and configures the specifications (see Table 2) that his/her design must fulfil. Each selected specification must configure the value of its associated W(s) and select the frequencies under which each specification must be verified. There is also a simultaneous generation, of associated bounds for each specification.
3) *Loop-shaping*. During this stage, the user performs the synthesis of the controller C(s) by shaping the open loop transfer function L0(s) in the Nichols diagram.
4) *Pre-filter design*. In this stage the user performs the synthesis of the pre-filter F(s) if the Type 6 2-DOF tracking specification has been previously activated by shaping of the minimum and maximum values of the closed loop transfer function in the Bode magnitude diagram.
5) *Validation*. In this stage the user makes sure that the specifications of his/her design are fulfilled.

When the user is working in a certain stage of the design, it is possible to advance on to the next stage or return to any of the previous stages. Readers who are interested can find a complete description of QFTIT in the User's Guide available at *http://ctb.dia.uned.es/asig/qftit/*

3.2 TIG

TIG assists the QFT designer in calculating template boundaries of interval plants, and plants with affine parametric uncertainty in their coefficients. These kinds of plants are very usual in control problems.

Let the following uncertain parameter vector be

$$p = [p_1, p_2, \ldots, p_L] \tag{11}$$

where pj $\in [p_j^{\min}, p_j^{\max}]$ j=1,2,..,L. With TIG it is possible to calculate the template boundaries of plants whose transfer functions have the following structure:

$$P(s;p) = \frac{b(s;p)}{a(s;p)} = \frac{\sum_{k=0}^{m} b_k(p)\cdot s^k}{\sum_{k=0}^{n} a_k(p)\cdot s^k} \tag{12}$$

where the coefficients $b_k(p)$ and $a_k(p)$ are linear combinations of the uncertain parameters, i.e.,

$$b_k(p) = \beta_{k0} + \sum_{j=1}^{L} \beta_{kj}\cdot p_j \tag{13}$$

$$a_k(p) = \alpha_{k0} + \sum_{j=1}^{L} \alpha_{kj}\cdot p_j \tag{14}$$

In the previous expressions β_{kj} and α_{kj} j=0,..,L are real constants. These kinds of plants are known as *plants with affine parametric uncertainty*.

A particular case of (12) is obtained when the plant transfer function coefficients are directly the uncertain parameters, such kinds of plants are known as *interval plants*:

$$P(s;q,r) = \frac{b(s;q)}{a(s;r)} = \frac{\sum_{k=0}^{m} q_k\cdot s^k}{\sum_{k=0}^{n} r_k\cdot s^k} \tag{15}$$

where $q_k \in [q_k^{\min}, q_k^{\max}]$ and $r_k \in [r_k^{\min}, r_k^{\max}]$.

For the plants described by (12) and (15), TIG includes four algorithms to calculate the associated templates. These algorithms are the following: a) the Bailey & Hui algorithm (Bailey and Hui, 1989), b) the Fu algorithm (Fu, 1990), c) the Kharitonov segment algorithm (Bartlet, 1993), and d) the grid algorithm.

Besides, TIG can also find the template boundaries associated with other kinds of plants whose templates have been previously computed in Matlab. Thus, TIG is able to cooperatively work with other software tools, like Matlab QFT Frequency Domain Control Design Toolbox (Borghesani et al., 1995) and QFTIT. In fact, TIG was developed to increase the kind of plants which can be used in QFTIT (the present version of QFTIT only works with plants expressed in RFF).

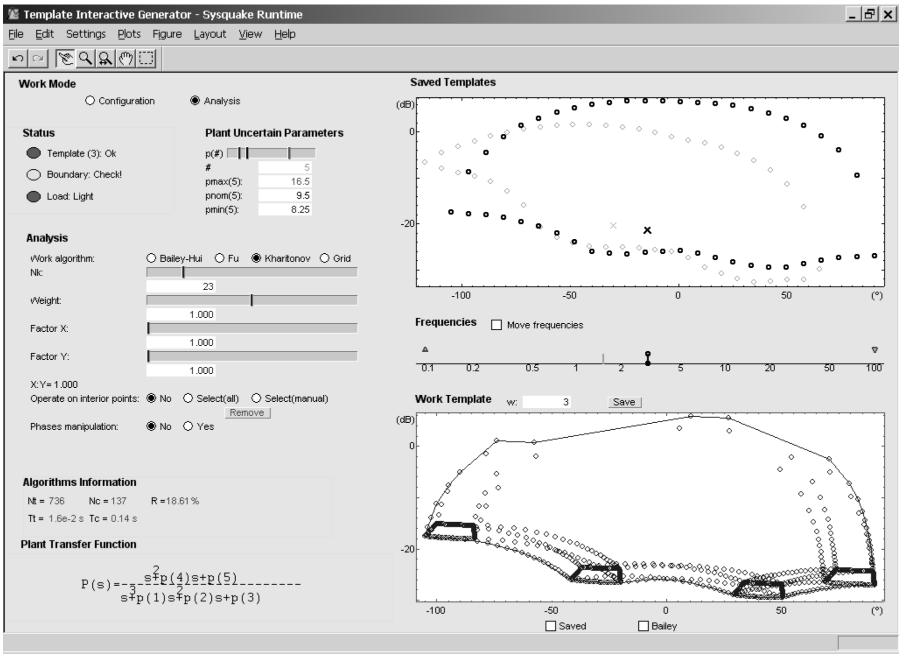


Fig. 3. Example of the TIG window in the Analysis mode

Like QFTIT, the main features of TIG are its ease of use and strong interactivity. These are common characteristics of all the software tools developed with the program SysQuake. Some examples of TIG interactivity are:

- The work template $\Gamma_w(\omega_w)$ (see Fig. 3.) in the Nichols chart (or on the complex plane) is simultaneously modified if users change the configuration parameter of the template computation algorithm or the work frequency ω_k.
- The boundary of the work template $\Gamma_w(\omega_w)$ in the Nichols chart (or on the complex plane) is simultaneously modified if users change the configuration parameter of the ε-algorithm (Montoya, 1998) used for computing the boundary template.

Other TIG features are:

- The simultaneous visualization of the work templates computed for each of the four template computation algorithms implemented in TIG.

- The automatic selection and removing of all $\Gamma_w(\omega_w)$ internal points.
- The manual selection and removing of any $\Gamma_w(\omega_w)$ point.

With TIG it is possible to work in two different modes: configuration and analysis. The configuration mode consists of four sequential steps:

- Step 1. Configuration of the plant uncertain parameters.
- Step 2. Configuration of the plant transfer function denominator.
- Step 3. Configuration of the plant transfer function numerator..
- Step 4. Configuration of the work frequencies set Ω.

Furthermore, in the analysis mode, users can do, among other things, the actions previously described. Readers who are interested can find a complete description in the User's Guide available at *http://ctb.dia.uned.es/asig/tig/*.

4 Illustrative Example: Stabilization of the Vertical Movement on a High-speed Ferry

4.2 Problem Statement

In a first approximation, the vertical acceleration of a high speed ferry is only associated with the pitch motion. Continuous linear models of the vertical dynamics of a high-speed ferry were identified (Aranda et al., 2004) for different navigation speeds (20, 30 and 40 knots). Based in this models, a family of plants $\mathcal{P}$ defined as a transfer function (output: pitch motion, input: position of the actuator (T-Foil)) with parametric uncertainties in its coefficients was obtained in (Díaz, 2002):

$$\mathcal{P} = \left\{ \begin{array}{ll} P(s) = \dfrac{K\left(s+a\right)\cdot\left(s+b\right)}{\left(s+103.2\right)\cdot\left(s+1.8\right)\cdot\left(s+c\right)\cdot\left(s^2+ds+e\right)} & \\ & \\ K \in \left[-0.87,-0.65\right] & a \in \left[-7.85,-6.67\right] \\ b \in \left[0.026, 0.042\right] & c \in \left[0.44, 0.49\right] \\ d \in \left[0.86, 0.97\right] & e \in \left[2.59, 2.80\right] \end{array} \right\} \tag{16}$$

The nominal plant is:

$$P_0(s) = \frac{-0.87(s-7.85)(s+0.042)}{(s+103.2)(s+1.8)(s^2+0.86s+2.8)} \tag{17}$$

A decrease in pitch motion is equivalent to reducing the system's sensitivity to the waves. From the point of view of frequency domain this means working with the sensitivity function S of the output (pitch) y to the perturbation (waves) d

$$S(s)=\frac{y(s)}{d(s)}=\frac{1}{1+P(s)\cdot C(s)} \tag{18}$$

A controller C must be designed, so that for all $P\in \mathcal{P}$ the system is stable

$$\left|\frac{C(j\omega)\cdot P(j\omega)}{1+C(j\omega)\cdot P(j\omega)}\right|\leq 1.2 \qquad \forall P\in\mathcal{P} \quad \forall\omega\in[0,\infty) \tag{19}$$

and for all disturbance $d\in D$ and frequency $\omega\in\Omega=[1, 2.5]$ (rad/s) the magnitude of the sensitivity function S is bounded by the specification

$$|S(j\omega)|\leq W_d(\omega) \qquad \forall P\in\mathcal{P} \quad \forall\omega\in[1{,}2.5] \tag{20}$$

where $W_d(\omega)$ is given in Table 1.

Table 1. Specification $W_d(\omega)$

ω [rad/sec]	$W_d(\omega)$ [dB]
1.00	-0.14
1.25	-0.21
1.50	-0.60
2.00	-0.75
2.50	-0.28

QFTIT will be used to solve this problem. Because the family of plants is given in RFF, it is not necessary to use TIG.

4.2 Stage 1: Template Computation

According to the disturbance rejection specification (see Table 1) and the robust stability specification (19), a possible set of trial frequencies is

$$\Omega_1 = \{1, 1.25, 1.5, 2, , 2.5, 10\} \quad (rad/s) \tag{21}$$

The frequencies Ω_1 can be introduced in QFTIT using its area *Template frequency vector* (see Fig. 2). There is a horizontal axis ω representing radian per seconds. It is possible to add, remove and change the frequencies of Ω_1. Each of these frequencies is represented by a vertical segment with an associated colour code that can be moved along the ω axis.

In this problem, the elements of the plant (16) are a gain, two simple zeroes, three simple poles and a pair of complex poles. One possible way of introducing them into QFTIT is using their areas *Operations over plant P* and *Uncertainty plant description*.

The area *Operations over plant P* is used to select the type of plant element (real-pole, real-zero, complex-pole, complex-zero, integrator) on which we want to perform some type of action (move, add or remove) in the *Uncertainty plant description* area. It is also possible to configure each element by using two sliders: the uncertainty of the delay and the gain of the plant, i.e. the specification of the minimum, maximum and nominal values. For the plant (16) the slider associated with the gain would have to be moved in order to configure its minimum value k_{min}= -0.87, its maximum value k_{max}=-0.65 and its nominal value k_{nom}=-0.87.

The *Uncertainty plant description* area is used to graphically design the configuration of the uncertainty of the plant poles and zeroes. This operation is carried out with the use of the mouse over the selected pole or zero element. For simple zeroes or poles the uncertainty is represented by a segment, whilst for complex zeroes and poles it is represented by a circular sector limited by the maximum and minimum values of the damping factor and the natural frequency of each complex item (pole or zero). Both representations include the extreme values as well as the nominal value.

For this problem, according to the plant defined in (16), by selecting the adequate options in the *Operations over plant P* area, it would be possible to add two simple zeroes (s=-a, s=-b), three simple poles (s=-103.2, s=-1.8, s=-c) and a pair of complex poles ($s = -d \pm j \cdot \sqrt{d^2 - 4 \cdot e}$) in the Uncertainty plant description area and to configure the uncertainty (a, b, c, d, e) of these elements and their nominal values (17) by dragging the mouse.

The area *Templates* shows a Nichols diagram that includes six templates calculated for the set of frequencies defined in Ω_1.

4.3 Stage 2: Specifications

In this problem there are two specifications: robust stability specification (19) (Type 1) and disturbance rejection at plant output (20) (Type 2). If

Type 1 specification is selected and activated in the *Specification type zone*, then it is possible to configure the value of the constant W_s by simply dragging the slider from value 1 to the desired value 1.2. Just under the slider there is a display showing the value of the gain margin ($GM \geq 1.8$) and the phase margin ($PM \geq 49.2°$) obtained. It is also possible to view simultaneously and interactively how the specification modulus is being modified in the Bode diagram and how the associated bounds change in the Nichols diagram.

The specification of disturbance rejection at plant output (Type 2) for this problem (see Table 1) is given as a vector whose components are the values that the specification must take in dB at different frequencies. These kinds of specifications are called Point-to-Point (PP) in QFTIT. Thus, the configuration of this specification given in Table 1 is as follows: first, it is necessary to select and activate the *Type 2* specification in the *Specification type zone*. Second, it is necessary to select the PP mode in the *W(s) frequency-domain* specification zone. The *W(s) magnitude specification* zone displays circles in different colours placed in the trial frequencies of the specification and with a value of 0 dB. Users can configure by dragging the mouse pointer over the modulus points to the chosen value. This will simultaneously update the bounds associated with this specification in the Nichols.

In QFTIT the final bounds $B(\omega_i)$ $\omega_i \in \Omega_1$ associated at the intersection of the two configured specifications are immediately displayed in the *Nichols Plot* zone by selecting the option *Intersection* in the *Option plot* zone.

4.4 Stage 3: Loop Shaping

During this stage, the user performs the synthesis of the controller $C(s)$ in the Nichols diagram by shaping the open loop transfer function L_0 in order to maintain the boundaries $B(\omega_i)$ $\omega_i \in \Omega_1$. The main manipulation that the user can perform within this area of the programme is the displacement of L_0 in certain directions depending on the selected controller item in the *Operations Over Controller C* zone.

The changes made to L_0 in the Nichols diagram are immediately reflected in an interactive way on the zeroes-poles map corresponding to C(s) as well as in the symbolic expression of the transfer function. Likewise, the interactions performed by the user on the zeroes-poles map of the controller will be reflected in the Nichols diagram. Thus, the user has a very interactive and flexible tool to perform the synthesis of the controller.

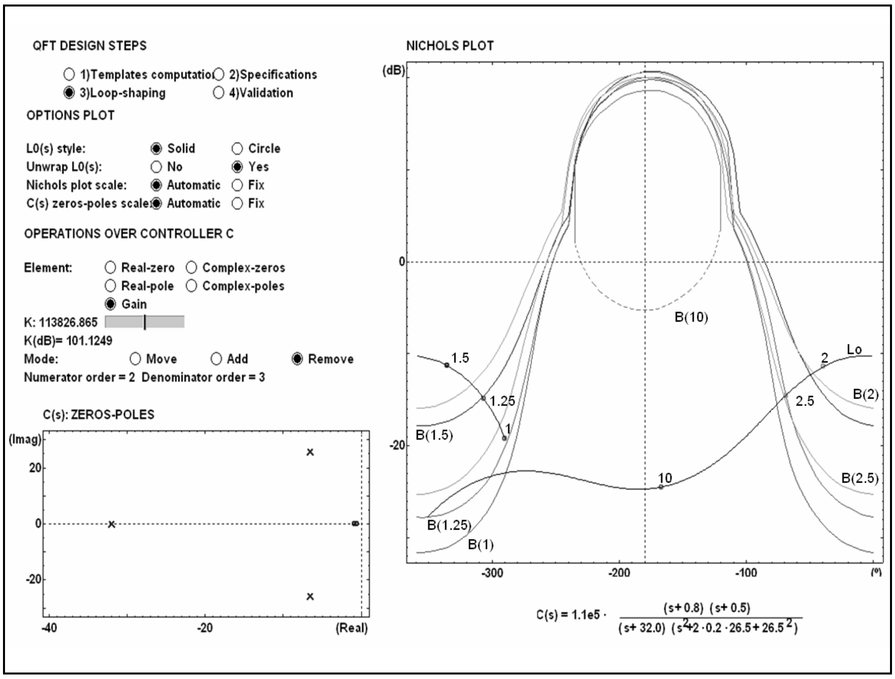


Fig. 4. Aspect of the QFTIT window after designing the controller *C*

Figure 4 displays the aspect of the QFTIT window after the Loop-Shaping stage. The Nichols diagram shows the intersection of the bounds associated with the established specifications and the final L_0. It can be observed how the points $L_0(j\omega_i)$ fulfil the boundaries $B(\omega_i)$ $\omega_i \in \Omega_1$.

The expression of the designed controller is:

$$C(s) = 1.1 \cdot 10^5 \cdot \frac{(s+0.5)\cdot(s+0.8)}{(s+32)\cdot\left(s^2 + 2\cdot 0.2\cdot 26.5\cdot s + 26.5^2\right)} \tag{22}$$

It is a controller with two real zeroes, one real pole and a pair of complex poles. The zeroes and the poles of $C(s)$ are represented in the *C(s): Zeros-Poles* zone.

4.5 Stage 4: Validation

During this stage designers make sure that the specifications of their design are fulfilled. The user only has to select the type of specification to validate, and QFTIT immediately shows the modulus of W_{si} and the worst case modulus of the associated characteristic function of the system in a Bode magnitude diagram.

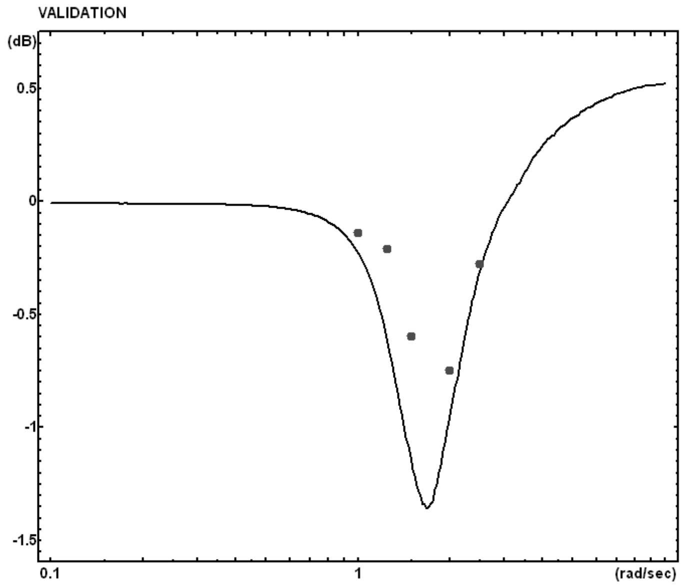


Fig. 5. Maximum magnitude in dB of the maximum magnitude of the sensitivity function $max\{|S(j\omega)|\}$ (solid line) and $W_d(\omega)$ (circles).

For this, two specifications have to be validated: disturbance rejection at plant output (20) and robust stability (19). Figure 5 displays $W_d(\omega)$ (circles) and the maximum magnitude of the sensitivity function $max\{|S(j\omega)|\}$ (solid line). It can be observed how the specification of disturbance rejection at plant output (20) is fulfilled, since $max\{|S(j\omega)|\}$ is below $W_d(\omega)$ in the design range of frequencies Ω=[1,2.5] (rad/s).

On the other hand, Figure 6 shows the maximum magnitude in dB of the closed-loop transfer function $max\{|L(j\omega)/(1+L(j\omega))|\}$ and the constant gain line W_s=1.2 (1.58 dB). As $max\{|L(j\omega)/(1+L(j\omega))|\}$ does not surpass the horizontal line in any of the frequencies, the robust stability specification would be correct with the controller C designed during step 3. This design assures a phase margin $PM \geq 50°$ and a gain margin $GM \geq 1.8$.

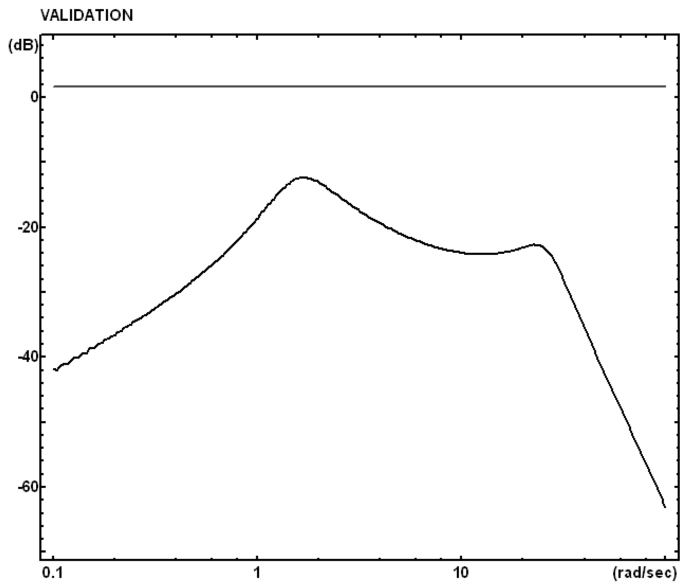


Fig. 6. Maximum magnitude in dB of the closed-loop transfer function $max\{|L(j\omega)/(1+L(j\omega))|\}$ and the constant gain line W_s=1.2 (1.58 dB).

Finally, time simulation was done using the controller (22) at four working points (ship speed U=30, 40 knots and sea state number SSN= 4, 5) The designed controller ensures a decrease around 10.9 % of the vertical acceleration associated to the pitch motion.

5 Conclusions

This paper has described the basic features of the software tools QFTIT and TIG for the design of robust controllers by means of the QFT methodology. The main features of these tools are its ease of use and its strong interactivity. Any action carried out on the screen by users is immediately reflected on all the graphs generated and displayed by the tool. This allows users to visually perceive the effects of their actions. Both of these tools are freely available as executable files for Windows or Mac platforms.

Besides, this paper has shown the utility of these tools for solving robust control problem in marine systems. A robust controller has been designed to stabilize the vertical movement of a high-speed ferry.

Acknowledgements

This development was supported by CICYT of Spain under contracts DPI2003-09745-C04-01 and DPI2004-01804.

References

Aranda, J., Cruz, J.M, Díaz, J.M., and Dormido Canto, S. 2002 QFT versus classical gain scheduling: study for a fast ferry. In Proc. 15th IFAC World Congress, Barcelona, Spain, pp. 331-336.

Aranda, J. Revilla, J., Díaz, J.M., and Ruipérez, P. 2002. Experimental and robustness analysis of a multivariable control for vertical dynamic of a high speed craft. In Proc. IEEE Int. Conf. on Control Applications, Glasgow, Scotland, pp. 1070-1075.

Aranda, J., Cruz, J.M., and Díaz, J.M. 2004. Identification of multivariable models of fast ferries. European Journal of Control. 10(2): 187-198.

Aranda, J., Muñoz, R., and Díaz, J.M., 2004. The problem of the vertical movement control with roll movement in fast ferries. In Proc. IFAC Control Applications in Marine Systems, Ancona, Italy, pp. 339-344.

Aranda, J., Cruz, J.M., and Díaz, J.M. 2005. Design of a multivariable robust controller to decrease the motion sickness incidence in fast ferries. Control Engineering Practice 13(8): 985-999.

Aranda, J., Muñoz, R, Díaz, J.M., and Dormido Canto, S. 2005. Robust control for the coupling of lateral and longitudinal dynamics in high-speed crafts. In Proc. 16 th IFAC World Congress, Prague.

Aranda, J, Chaos, D., Dormido-Canto, S., Muñoz, R., and Díaz, J.M. 2006. Benchmark Control problems for a non-Linear underactuated hovercraft: a simulation laboratory for control testing. In Proc. Advances in Control Education, Madrid, Spain.

Aranda J., Chaos, D, Dormido-Canto, S., Muñoz, R., Díaz, J.M. 2006. A Control for tracking and stabilization o fan underactuated non-linear RC hovercraft. In Proc. Maneuvering and Control of Marine Systens. Lisbon, Portugal.

Åström, K.J. 1994. The future of control. Modeling, Identification and Control 1994 15(3): 127-134.

Åström, K.J. 2000. Limitations on control system performance. European Journal of Control 6(1): 2–20.

Bailey, F. N., and Hui, C. H. 1989. A fast algorithm for computing parametric rational functions. IEEE transactions on automatic control 34(11) 11: 1209-1212.

Bartlett, A. C., Tesi, A., and Vicino, A. 1993. Frequency response of uncertain systems with interval plants. IEEE transactions on automatic control 38(6): 929-933.

Borghesani C., Chait Y, and Yaniv, O. 1995. Quantitative Feedback Theory Toolbox - for use with MATLAB. The MathWorks Inc, Natick, MA.

Cruz J.M, Aranda, J., Giron-Sierra, J.M, Velasco, F., Esteban, S., Diaz, J.M., and Andres-Toro, B. 2004. Improving the Comfort of a Fast Ferry, Smoothing a ship's vertical motion with the control of flaps and T-foil. IEEE Control System Magazine 24(2): 47-60.

Diaz, J.M. 2002. Identificaión, modelado y control de la dinámica vertical de un buque de alta velocidad. Ph. D. Thesis, Universidad Nacional de Educación a Distancia (UNED), Madrid, Spain. Dpto. de Informática y Automática.

Díaz, J. M., Dormido, S., and Aranda, J. 2005. Interactive computer-aided control design using Quantitative Feedback Theory: The problem of vertical movement stabilization on high-speed ferry, International Journal of Control 78(11): 793-805.

Díaz, J. M., Dormido, S, and Aranda, J. 2005. QFT robust control design by interactive approach, In Proc. 44th IEEE Conference on Decision and Control, and European Control Conference, Sevilla, Spain, pp. 1301-1006.

Dormido, S. 2003. The role of interactivity in control learning (Plenary Lecture). In Proc. 6th IFAC Symposium on Advances in Control Education, Oulu, Findland, pp. 1-12.

Dormido, S. 2004. Control learning: Present and future. Annual Reviews in Control 28: 115-136.

Fu, M. 1990. Computing the frequency response of linear systems with parametric perturbation. Systems & Control Letters 15: 45-52.

Gutman, P.O., Baril, C., and Neumann, L. 1995. An algorithm for computing value sets of uncertain transfer functions in factored real form. IEEE Transactions on Automatic Control 29(6):1268-1273.

Horowitz, I. M. 1963. Synthesis of feedback systems. Academic Press: New York.

Horowitz, I. M. 1992. Quantitative feedback design theory (QFT). QFT Publishers 660 South Monaco Dorkway Denver, Colorado.

Horowitz, I. M. 2001. Survey of Quantitative Feedback Theory (QFT), International Journal of Robust and Non-linear Control 11(10): 887-921.

Houpis, C. H., Rasmussen, S. J., and García-Sanz M. 2006. Quantitative Feedback Theory: fundamentals and applications, 2nd Edition, CRC Taylor & Francis: Boca Ralen.

Johansson, M., Gäfvert, M.,, and Åström, K.J. 1998. Interactive tools for education in automatic control. IEEE Control Systems Magazine 18(3): 33-40.

Montoya, F. J. 1998. Diseño de sistemas de control no lineales mediante QFT: Análisis computacional y desarrollo de una herramienta CACSD. Ph. D. Thesis. Universidad de Murcia (Spain), Departamento de Informática y Sistemas Informáticos.

Muñoz, R., Aranda, J., Díaz, J. M., Dormido-Canto, S., and Chaos, D. 2006. Robust control design by QFT methodology for dynamic positioning problem of a moored floating platform. In Proc. Maneuvering and Control of Marine Systens. Lisbon, Portugal.

Piguet, Y. 1999. Sysquake user manual, version 1.0, Calerga: Lausanne, Switzerland.

Wittenmark, B., Häglund, H., and Johansson, M. 1998. Dynamic pictures and interactive learning. IEEE Control Systems Magazine 18(3): 26-32.

Yaniv, O. 1999. Quantitative feedback design of linear and nonlinear control systems. Kluwer Academic Publishers: Norwell, Massachusetts.

CHAPTER 3

Load disturbances compensation techniques on marine propulsion systems

R. FERREIRO GARCIA[1] and M. HARO CASADO[2]

[1]ETSN y Máquinas, Universidade da Coruña, ferreiro@udc.es;
[2] Facultad de Ciencias Náuticas, Universidad de Cádiz,
manuel.haro@uca.es

This chapter describes the advances achieved of a research work dealing with the disturbances rejection on an engine propulsion system under propeller *in-and-out water* effect due to severe weather conditions. Proposed strategy consists in a cascade structure based on a feedback controller, which commands a reference model, followed by an embedded backstepping algorithm that takes the reference signals from the reference model. With this procedure the conventional governor problems regarding disturbances rejection are efficiently overcome.

1 Introduction

When under heavy weather conditions, mainly on the bow-astern direction, large pitch motion amplitudes are generated, thus producing high propulsion moving parts acceleration associated to high and dangerous inertia loads on main engine. Such abrupt changes on engines acceleration are due to changes on propeller torque as consequence of alternative vertical motion of the ship and propeller, mainly when pitch motion is severe and cargo condition is ballast. The acceleration of some moving parts of propulsion system must be limited in order to avoid overspeed as well as surpassing some maximum allowable inertia loads.

Modelling of thrust loads, which are constituted of complex physical phenomena, is most commonly solved by empirical methods in conjunction with analytical models, see e.g. Minsaas *et al.* (1987); Sørensen *et al.* (1997); Smogeli *et al.* (2004), Smogeli *et al.* (2005), and references

therein. The most important loss effects that affect the propeller torque/power dynamics are:
1. *Fluctuations* in the in-line water inflow to the propeller, where a constant shaft speed and *varying advance velocity* will lead to varying thrust and torque.
2. *Ventilation* (air suction) caused by decreasing pressure on the propeller blades for heavily loaded propellers, especially when the effective submergence of the propeller becomes small due to waves and the wave frequency motion of the vessel.
3. *In-and-out-of-water effects* or propeller emergence due to the relative propeller motion that results in water exits. The above-mentioned disturbance is the most relevant responsible for over-acceleration, which must be limited by means of an effective control strategy.

The conventional governors based on PI(D_F) (D_F means low pass filtered derivative mode) modes contributes with an inherent phase lag, thus avoiding a fast reaction under abrupt load changes. Conventionally, such problems are overcome by means of a feedforward control structure. However, in this case this strategy is not effective because load changes can not be measured accurately.

Such is the main motivation to improve the propulsion engine dynamics safety by proposing, designing and applying a state of the art controller achieved by means of the combination of a feedback PID controller followed by an embedded well-known backstepping procedure.

Since this research work depends on a model-based approach, a dynamic model of the propulsion system is to be first achieved. The task of modelling the propulsion system is carried out in next section. Section 3 describes the backstepping procedure. Section 4 describes the implementation of the backstepping procedure applied to the propulsion system model. Section 5 deals with the strategy based on the integration of the backstepping procedure into the feedback control loop, and finally section 6 deals with the discussion of results and conclusions.

2 Modelling the Propulsion System

The speed control of a diesel engine based propulsion system is shown in figure 1. In this scheme it is shown that the angular speed n of the shaft is being controlled conventionally by means of a feedback PID controller (The governor).

The load on the engine shaft is due to the inertial loads inherent to rotating masses I_E, to the friction resistances and propeller torque Q_P. The ship dynamics depends on propeller thrust T_P, hull viscous resistances and disturbances.

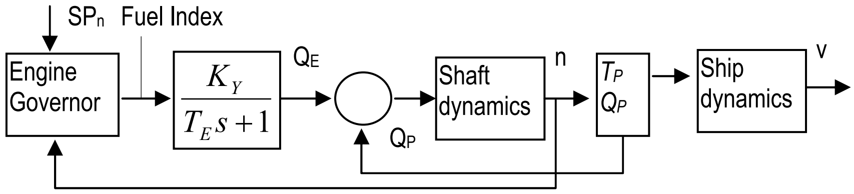


Fig. 1. Diesel engine based propulsion scheme

2.1 Engine Dynamics

In the task of modelling the main engine dynamics it is assumed that the propeller shaft is driven by a Diesel engine, which generates a torque Q_E controlled by a fuel index *Y,* Blanke, M., K *et al*. (2000, 2003) and Izadi *et al.* (1998, 1999). The Diesel engine dynamics is splitted into two sub-models. The first part describes the relation between the developed torque and the fuel index. The linear transfer function is modelled as a first order linear time invariant system according

$$\frac{Q_E}{Y} = \frac{K_Y}{T_E s + 1} \tag{1}$$

where K_Y is the gain constant and T_E is the time constant corresponding to the torque from cylinder firings. The mentioned second part correspond to the shaft torque balance, being expressed as

$$Q_E = I_E \dot{n} + Q_F + Q_p \tag{2}$$

where I_E is the total moment of inertia of rotating parts plus added inertia, Q_p is the torque developed from the propeller and Q_F is the shaft friction torque which depends on the shaft speed.

Some dynamic characteristics for a specific Diesel Engine are given by MAN B&W (2006) in steady state conditions. The relationship between such characteristics is shown in table 1. This tale shows the load in (%), the power developed (kW), the speed in (rpm), the specific fuel oil consumption (SFOC) in grams per kW and hour, and the power absorbed by mechanical friction of rotating parts in (kW). Some useful relations can be achieved from the data of table 1 in order to model the engine dynamics.

Table 1. Characteristics of a specific Diesel Engine

load %	Power kW	speed r/min	SFOC g/ kWh	Friction kW
100	20,090	95.0	182,800	1,430
95	19,086	93.4	175,900	1,410
90	18,081	91.7	169,000	1,390
85	17,076	90.0	162,000	1,370
80	16,072	88.2	154,700	1,340
75	15,068	86.3	147,200	1,310
70	14,063	84.4	139,400	1,280
65	13,058	82.3	131,300	1,240
60	12,054	80.1	122,800	1,200
55	11,050	77.8	114,000	1,160
50	10,045	75.4	104,800	1,110
45	9,040	72.8	95,300	1,060
40	8,036	70.0	85,400	1,010
35	7,032	66.9	75,200	950
30	6,027	63.6	64,800	880
25	5,022	59.8	54,100	810

According the data provided by table 1, with the help of a well known functional approximation technique based on the polynomial regression to fit the data to the following equations with a verified goodness greater than 0.994, the following relations are achieved:

$$Friction\_Power(kW) = -865.65 + 2.0445 \cdot n - 372.2425 \cdot n^2 \quad (3)$$

$$Friction\_Torque(kW / n) = \frac{Friction\_power(kW)}{\omega} = \frac{-865.65 + 2.0445 * n - 372.2425 * n^2}{\omega} \quad (4)$$

$$Brake\_Power(kW) = 10.6782E3 - 25.26E3 \cdot n + 19.6921E3 \cdot n^2 \quad (5)$$

$$Brake\_Torque(kW / \omega) = \frac{10.6782E3 - 25.26E3 \cdot n - 19.6921E3 \cdot n^2}{\omega} \quad (6)$$

$$Fuel\_Flow(kg / s) = 2.3834 - 4.767 \cdot n + 2.4424 \cdot n^2$$

$$Power(kW) = 3.737E3 + 22.8976E3 \cdot FF - 6.9407E3 \cdot FF^2 \quad (7)$$

where FF is the fuel flow.

$$Q_E(kW / n) = \frac{3.737E3 + 22.8976E3 \cdot FF - 6.9407E3 \cdot FF^{2^{\,2}}}{\omega} \quad (8)$$

with ω the angular speed of the propeller shaft.

2.2 Propeller Thrust Dynamics

The following relations model the propeller thrust and torque

$$\begin{aligned} T_p &= K_T \rho D^4 |n| n \\ Q_p &= K_Q \rho D^5 |n| n \end{aligned} \tag{9}$$

where D is the propeller diameter and ρ is the mass density of water. K_T and K_Q are respectively torque coefficient and thrust coefficient which are defined by Pivano *et al.* (2006a)

$$K_T = \frac{T_p}{0.5 \cdot \rho \cdot V_r^2 \cdot A_0} \tag{10}$$

$$K_Q = \frac{Q_p}{0.5 \cdot \rho \cdot V_r^2 \cdot A_0 \cdot D} \tag{11}$$

where V_r is the relative speed of advance and A_0 is the propeller disc surface. The relative speed can be defined by Pivano *et al.* (2006b)

$$V_r^2 = V_a^2 + (0.7 \cdot R \cdot n)^2 \tag{12}$$

where R is the propeller disc radius and V_a is the speed of advance (arriving water velocity to propeller)
The corresponding angle between both velocities is achieved from (12) as

$$\beta = a\tan(V_a, 0.7 \cdot R \cdot n) \tag{13}$$

where the advance velocity can be achieved from the advance number J as

$$J = \frac{V_a}{n \cdot D} \tag{14}$$

As V_a is variable according the type of propeller and the dynamic status, then J is also variable according those conditions.

2.3 Ship Surge Dynamics

The ship dynamics can be approached by the following nonlinear differential equation

$$m\dot{v} = R(v) + (1 - t_T)T_p + T_{EXT} \tag{15}$$

where m is the total mass (ship mass plus added mass), $R(v)$ is the hydrodynamic resistance, $(1\text{-}t_T)$ is the thrust deduction factor, T_{EXT} is the total external forces and Tp is the thrust propulsion.

2.4 The Nonlinear State Space Model

After rearranging eq.(1-3), and neglecting the external forces, engine speed and ship velocity is achieved a block diagram is achieved and shown in figure 2.

$$\begin{aligned}\dot{n} &= [Q_E - Q_F(n) - Q_p]/I_E \\ &= \frac{1}{I_E}[Q_E - Q_F(n) - K_Q \cdot \rho \cdot D^5 \cdot n \cdot |n|]\end{aligned} \tag{16}$$

$$\begin{aligned}\dot{v} &= [-R(v) + (1-t_T)T_p]/m \\ &= \frac{1}{m}[-R(v) + (1-t_T) \cdot K_T \cdot \rho \cdot D^4 \cdot n|n|]\end{aligned} \tag{17}$$

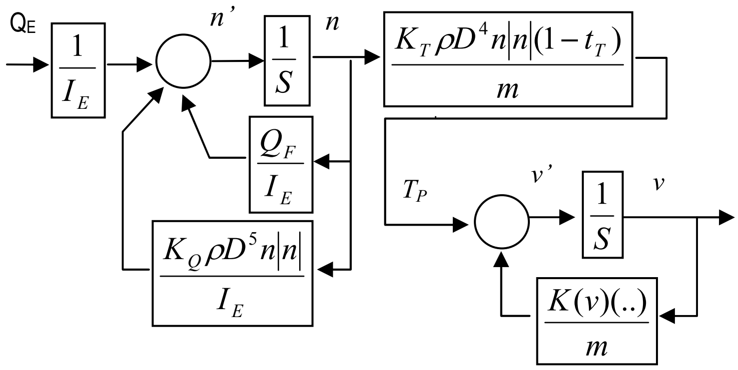


Fig. 2. The propulsion system block diagram.

With regard to torque disturbances, the *in-and out-of-water* effect refers to a static loss of thrust when the propeller is partially submerged. In such a case the torque loss can be directly related to the lost of effective disc area, as given by Minsas *et al*. (1987). However, under severe weather conditions any attempt to achieve a model of such phenomena don't success at all. Consequently, the disturbances coming from propeller torque due to all mentioned prejudicial effects would be measured by computing shaft acceleration. In order to model the effect of such disturbances, (14) is corrected with a parameter C_D that will take any value into the range $0.3 < C_D < 1$. As consequence of such correction equation (16) yields

$$\dot{n} = \frac{1}{I_E}[Q_E - Q_F(n) - C_D \cdot K_Q \cdot \rho \cdot D^5 \cdot n \cdot |n|] \tag{18}$$

where the dynamics of C_D is associated with the pitch motion characteristics (frequency and amplitude). The shaft resistances are also complex dynamic functions and are estimated with experimental data from MAN B&W (2006), which are restricted to a unique propulsion plant type.

3 The Backstepping Procedure

Nonlinear control theory has been the subject of very strong devolvement during the last two decades. Control systems have one main goal to achieve, and that is the stability of the controlled system. There are different kinds of stability problems, which occur when studying dynamical systems. The tools developed in this area suddenly made the design and implementation of controlling units in nonlinear systems more structured and rather straightforward. One of the concepts, which are well known today, is Backstepping theory. The Backstepping theory provides a tool for the recursive design of the control law based on the Lyapunov theory for stabilizing the controlled system. This work concerns with stability of equilibrium points LaSalle, J.P (1966) and Yoshizawa, T. (1968). This section briefly reviews Control Lyapunov Functions and Backstepping theory and formalizes this requirement based mainly in Krstic M., I. Kanellakopoulos, and P. Kokotovic. (1995) and Krstic M. and H. Deng (1998).

3.1 Control Lyapunov Functions (CLF)

Since an objective is to develop closed-loop systems with desirably stability properties, let us consider a system excited with a control input u

$$\dot{x} = f(x,u),\ x \in R^n,\ u \in R,\ f(0,0) = 0 \tag{19}$$

It is necessary to introduce an extension of the Lyapunov function concept, the CLF. The objective is to find a control law $u = \alpha(x)$ such that the desired state of the closed-loop system

$$\dot{x} = f(x, \alpha(x)) \tag{20}$$

is a GAS (globally asymptotically stable) equilibrium point. It is considered the origin to be the goal state for simplicity. It can be chosen a function $V(x)$ as a Lyapunov candidate, and require that its derivative along the solutions of (19) satisfy $\dot{V}(x) \le -W(x)$, where $W(x)$ is positive definite function. Then closed-loop stability follows from a theorem. It is necessary therefore to find $\alpha(x)$ to guarantee that for all $x \in R^n$

$$\dot{V}(x) = \frac{\partial V}{\partial x}(x) f(x, \alpha(x)) \leq -W(x) \qquad (21)$$

The pair V and W must be chosen carefully otherwise (20) will not be solvable Krstic M. and H. Deng (1998). This motivates the following definition, which can be found in .

Definition 1 (CLF)

A smooth positive definite and radially unbounded function $V: R^n \rightarrow R_+$ *is a* CLF *if for(19) and every* $x \neq 0$

$$\dot{V}(x) = V_x(x) f(x,u) < 0, \ \forall x \neq 0 \text{ and for some } u \qquad (22)$$

The significance of this definition concerns to the fact that, the existence of a globally asymptotic stabilizing control law is equivalent to the existence of a CLF. If exists a CLF for the system, then it is certainly possible to find a globally stabilizing control law. The reverse is also true. This is known as Artestin's theorem and can be found in Sontag E.D., (1989). Now that the concept CLF is defined, it is necessary to move on and explore the backstepping theory, which is the main tool we must deal with in order to design a tracking control law.

3.2 Backstepping

The main drawback of the CLF concept as a design tool is that for most nonlinear systems a CLF is not known. The task of finding an appropriate CLF may be complex as well as that of designing a stabilizing feedback law. The backstepping procedure solves both problems simultaneously. The standard procedure can be found in Krstic M., I. Kanellakopoulos, and P. Kokotovic. (1995), Krstic M. and H. Deng (1998), Fosen, T.I. and A. Grovlen (1998) and Fosen, T.I. and J.P. Strand (1999). To spare the reader from the labor of understanding the main ideas of backstepping by a theorem, it is preferable to start from the description of the general procedure, which will clarify such concepts.

3.2.1 General Backstepping Procedure

Backstepping consists in a structured sequence of steps responsible for stabilizing a general nonlinear process control described into a strict feedback state space form. Given a SISO nonlinear process described under a state space model, and assuming Y_r to be a tracking reference trajectory, the steps in the next procedure are to be performed under a systematic way according the following sequence:

Step 1:
Error step 1:

$$Z_1 = Y_r - X_1$$

Lyapunov Function 1st step:

$$V_1 = \frac{1}{2} Z_1{}^2$$

Derivative of V_1:

$$\dot{V}_1 = Z_1 \dot{Z}_1 = Z_1 (\dot{Y}_r - X_2)$$

Stabilizing function 1st step:

$$\alpha_1 = X_{2d} = \dot{Y}_r - P_1 Z_1$$

Step 2:
Error step 2:

$$Z_2 = X_2 - X_{2d} = X_2 - \alpha_1$$

Lyapunov Function 2d step:

$$V_2 = \frac{1}{2} Z_1{}^2 + \frac{1}{2} Z_2{}^2$$

with $\dot{Z}_1 = \dot{Y}_r - \dot{X}_1 = \dot{Y}_r - X_2 = P_1 Z_1 - Z_2$

Derivative of V_2:

$$\dot{V}_2 = P_1 Z_1{}^2 - Z_2 (Z_1 + \dot{X}_2 - \dot{\alpha}_1)$$
$$= P_1 Z_1{}^2 - Z_2 \left[(1 - P_1^2) Z_1 + P_1 Z_2 + X_3 - \ddot{Y}_r \right]$$

Stabilizing function 2d step:

$$\alpha_2 = X_{3d} = (P_1^2 - 1) Z_1 - (P_1 + P_2) Z_2 + \ddot{Y}_r$$

Step i:
Error step *i:*

$$Z_i = X_i - X_{id} = X_i - \alpha_{i-1}$$

Lyapunov Function *i*th step:

$$V_i = \frac{1}{2} \sum_{j=1}^{i} Z_j{}^2$$

with $\dot{Z}_{i-1} = Z_i - P_{i-1} Z_{i-1} - Z_{i-2}$

Derivative of V_i:

$$\dot{V}_i = -\sum_{j=1}^{i-1} P_j Z_j{}^2 + Z_i \left[Z_{i-1} + \dot{X}_i - \dot{\alpha}_{i-1} \right]$$

Stabilizing function i^{th} step:

$$\alpha_i = X_{i+1} = -P_i Z_i - Z_{i-1} + \dot{\alpha}_{i-1}$$

where $P_i > 0$, are positive constants

Up to now, a number of stabilizing functions have been achieved. In the last step a feedback control function is defined. The above procedure will be applied on the determination of the parameters for a tracking controller (governor) of the propulsion engine in order to reject severe disturbances mainly due to *in-and-out* of water propeller.

4 Tracking Controller Design Procedure

Among the most relevant features of a marine propulsion system are those regarding the Digital Governor System (DGS) and the Main Engine Safety System (MESS).

The DGS performs operations such

- Regulation of main engine speed by advanced regulating algorithms.
- Load controller "scavenge air", "torque", "manual" load limiter.

The MESS is responsible for

- Automatic shut down and emergency stop.
- Automatic slow down system.
- Over speed protection system using dual engine speed detection.

Among mentioned safety systems operations are those regarding overspeed, which are rarely used except when a scenario composed by severe weather conditions (extreme pitch motions) under ballast cargo conditions is given and /or cavitation conditions etc. Such disturbances are directly related with shaft torque by means of equation (18) and cannot be effectively measured. An effective way to measure such disturbances consists in measuring the shaft torque by means of a torsional principle, which will transfer its output signal by means of a wireless device and is expensive.

4.1 Backstepping Implementation

From the scheme depicted with figure 2 representing the system dynamics, a state variable representation is shown in figure 3, where the ship dynamics is neglected because the influence of thrust changes is transferred to the torque, which affects directly the engine dynamics. Additionally, a disturbance input C_D is applied to the propeller torque in order to describe and simulate the *in-and-out of water* effect under severe weather scenarios responsible for generate large pitch motion amplitudes. In the above process model, the input is the engine torque developed by the cylinder firings. In the control algorithm to be achieved, such signal is the desired engine torque, and is generated by the controller output.

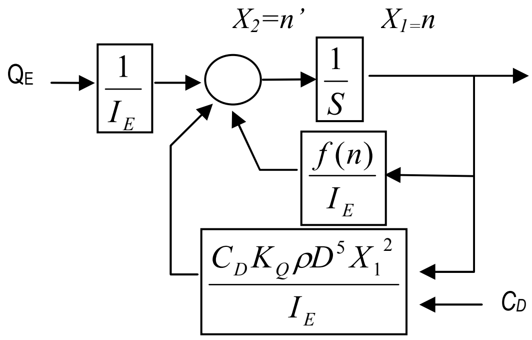


Fig. 3. State space model of disturbed engine

In the scheme of figure 3 it is assumed that the reference trajectory (reference speed) Y_r to be tracked is the engine speed $n = X_1$. Then the error dynamics is defined as

$$Z_1 = n_r - n = Y_r - X_1 \tag{23}$$

Following the described methodology, choose a candidate Lyapunov function such that

$$V_1 = \frac{1}{2} Z_1^{\ 2} \tag{24}$$

then the derivative of the Lyapunov function after introduce the derivative of eq(23) yields

$$\dot{V}_1 = Z_1 \dot{Z}_1 = Z_1(\dot{Y}_r - X_2) \tag{25}$$

To guarantee the stability, the derivative of the Lyapunov function is forced to a constant value defined by

$$\dot{V}_1 = -P_1 \cdot Z_1^{\ 2}; P_1 \geq 0 \tag{26}$$

where P_1 is a constant, which for this algorithm will take a value sufficiently high (20-30). Consequently, the above mathematical operations carried out with (25) and (26) yields

$$X_2 = \dot{Y}_r + P_1 \cdot Z_1 \tag{27}$$

Including in (27) the state space model depicted in figure 3 to achieve the manipulated variable or desired engine torque Q_{ED} yields the following stabilizing control law

$$Q_{ED} = -I_E\left(\frac{\dot{n} - \dot{Y}_r}{P_1} - Y_r\right) + Q_F + C_D \cdot K_Q \cdot \rho \cdot D^5 \cdot n^2 \qquad (28)$$

Equation (28) will attenuate the disturbances coming from propeller torque assuring a stabilizing control law.
The achieved control law will track a selected reference input, which will be generated via a reference model as shown in figure 4, where the feedback controller commands the reference model and the output of reference model is transferred to the backstepping algorithm.

5 Control Structure Implementation

The proposed structure has been designed to prevent propulsion system overspeed due to load disturbances from *in-and-out* of water effect on the propeller when severe weather conditions are given associated to severe pitch motions.

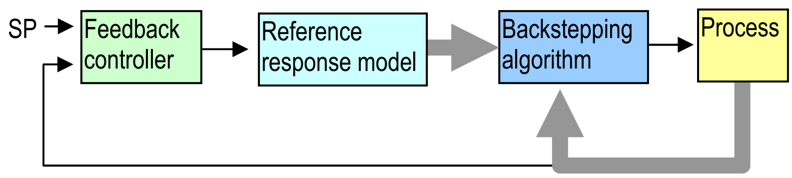


Fig.4. Structure of proposed strategy based on a cascade of two controllers.

The closed loop control structure is implemented by means of a feedback PID controller that commands a reference model under the structure shown in figure 4. Then, the reference model must provide a response that will be tracked by the backstepping controller. The feedback PID controller, which performs the role of a master controller, is characterized by a high proportional gain, which ensures a rapid proportional action to activate the reference model response, influencing the backstepping algorithm responsible for the command of the fuel rack index. Consequently, the high gain of the feedback controller operates as a detector of the load disturbances changes. As the high gain affect of the proportional action is transferred to the reference model, inherently is attenuated because the reference model is a low pass filter. Nevretheless, the load change event has been transmitted to the backstepping algorithm. Mentioned high proportional gain contributes with an unstable effect to the closed loop dynamics of the controlled system. However, this unstable effect is effectively compensated by means of the backstepping algorithm embedded in the loop.

The following features are inherent to the proposed control structure:

- Load disturbances coming from abrupt changes of propeller torque are difficult to be accurately measured. The unique event that prove or evidence load changes, is the changes of shaft speed under a given fuel index. That is the reason why a feedforward control will not be effective.
- The reference inputs to the backstepping algorithm from the reference model contribute to a rapid reaction of the backstepping algorithm when a type zero first order model describes the system reference model response.
- The fact by which the integral action of the master feedback controller is not an obstacle to be aware of load changes is due to the high proportional gain of this controller.
- The constant P_1 selected to the stabilizing function of the backstepping algorithm must be also high (20-25) in order to ensure an accurate model reference response tracking capacity.

The reference model is responsible for the generation of the desired speed and acceleration. By means of the reference model, the limits of allowable or desired controlled variables are defined. Figure 5 illustrates the location of avery contriol block into the closed loop control system.

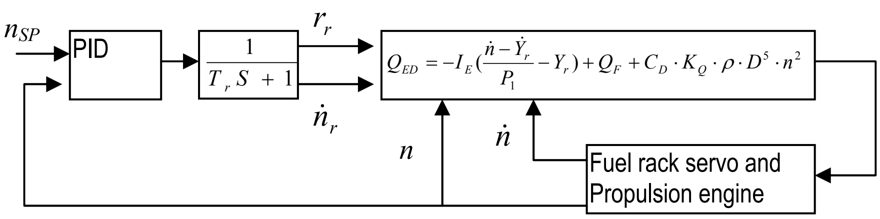


Fig. 5. Proposed closed loop control system

A simple model of the form

$$\frac{Y_r}{u} = \frac{n_r}{u} = \frac{1}{T_r s + 1} \tag{29}$$

is sufficient to define the desired speed and acceleration under the input command or control variable u. The desired acceleration (the maximum allowable acceleration) is defined by the time constant T_r according the expression achieved from (29)

$$\dot{n}_r = \frac{u - n_r}{T_r} \tag{30}$$

The closed loop control system after adjusting the master feedback controller and the basckstepping algorithm adopt the parameters shown in table 2.

Table 2 Contrl system parameters

Master PID	Reference model	Backstepping
Kp=8,Ti=40;Td=10,Tn=0.2	T_r=5	P_1=25

6 Results and Conclusions

In order to verify the effectiveness of the proposed control strategy, two case studies were considered on the same marine propulsion system: The case of a conventional governor based on a feedback PID controller and the case of an embedded backstepping algorithm into a closed loop feedback master controller according the control scheme depicted with figure 4.

Figures 6 and 7 show the results of both dynamic responses under the same sea state conditions that are under the same load disturbances. The *in-and-out of water* effect supposes a load variation between 40% and 110% of nominal load. Under the conditions of the described situation both propulsion systems were simulated. Every figure displays three graphs each. The first graph represents the setpoint speed in (rps) and the actual revolutions. The second graph shows the speed and acceleration of the reference model. The third graph shows the manipulated variable generated by the backstepping algorithm and the disturbances.

Comparing the first graph of figure 6 with the first graph of figure 7, a relevant difference exists with regard to the engine speed responses under disturbances. The results of figure 7 are much better and satisfactory than those of figure 6. In a real situation the results of figure 6 could not be tolerated because of the safety requirements. The actual conventional solution consists of reducing the engine speed setpoint in order to keep a reasonable degree of safety. Nevertheless, with the proposed solution such reduction is not necessary because control performance fulfil the safety requirements.

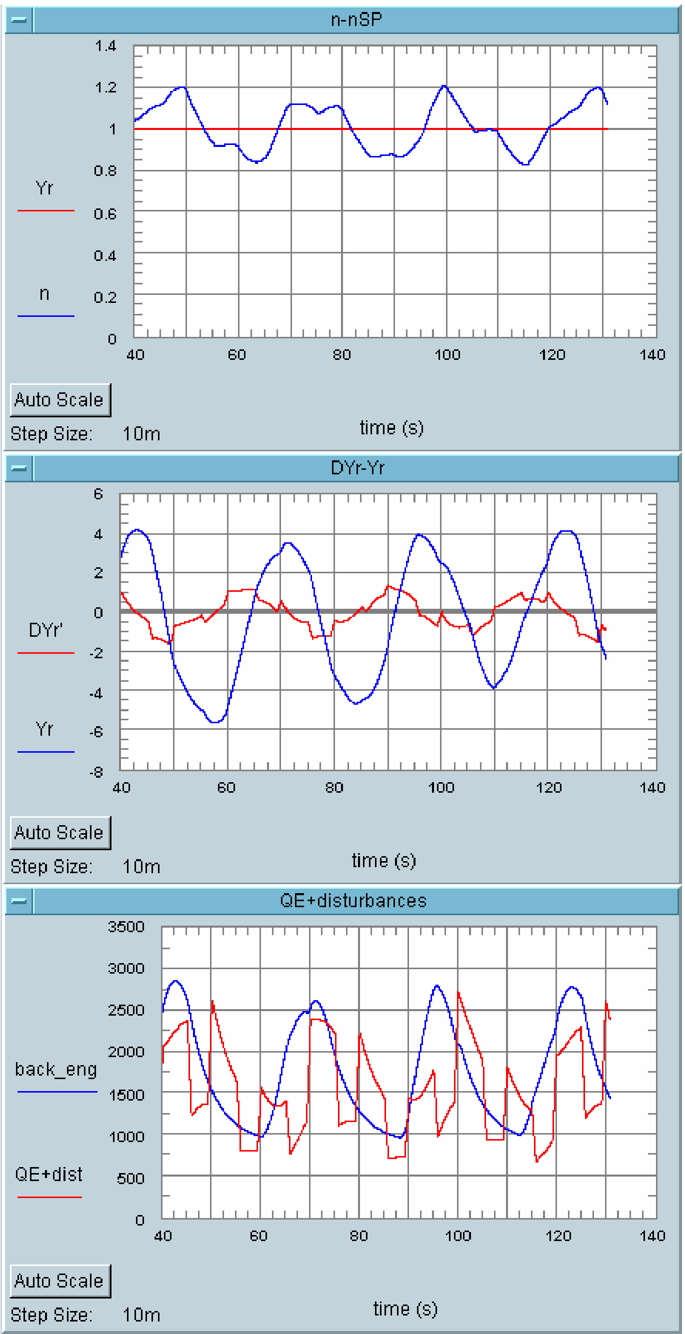


Fig. 6. Conventional propulsion system response

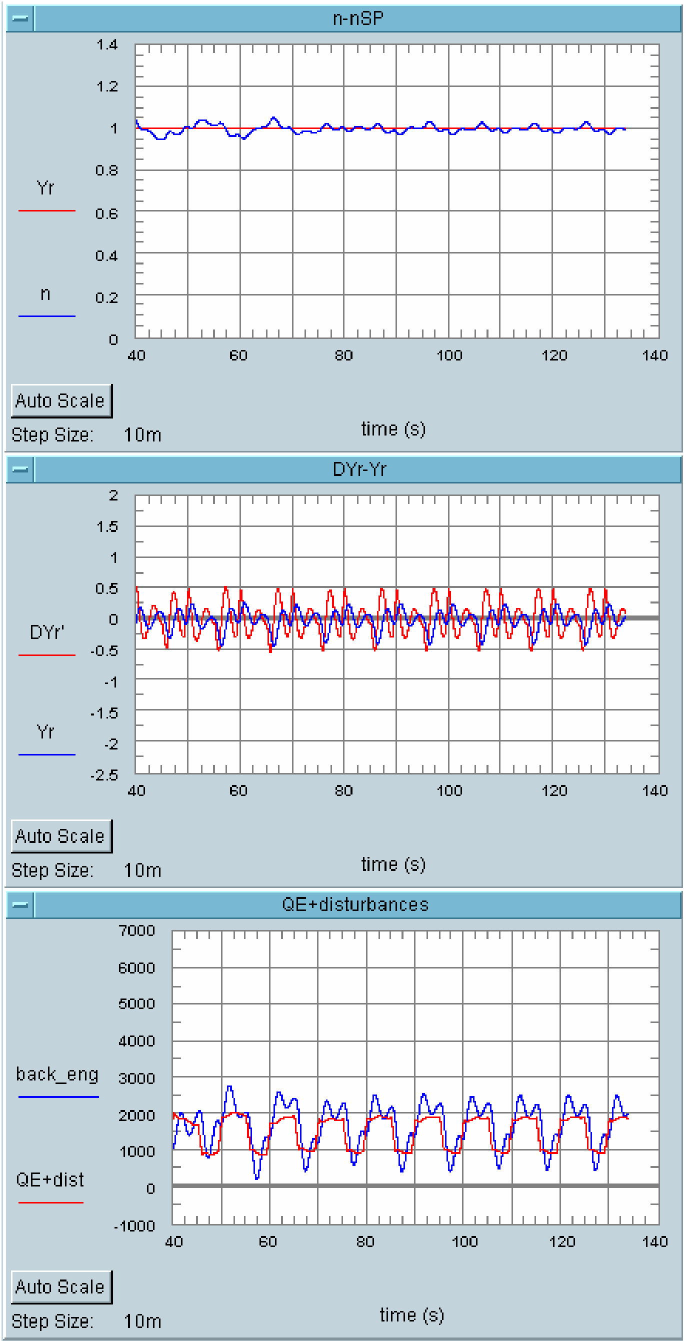


Fig.7. Proposed propulsion system response

6.1 Conclusions

The motivation of this work is the lack of performance of conventional governors applied on marine diesel propulsion systems under severe weather conditions. The main reason of such problem is the phase lag introduced by the integral action of the conventional (PID) governors, which is an obstacle to provide a rapid response under load changes. The mentioned unavoidable phase lag cannot be eliminated from a conventional governor, and this is the main reason by which this regulator is not quite effective actually under severe weather conditions. Another problem comes from the unstable effect introduced by increasing the proportional action gain in order to achieve a rapid response to load variations.

To overcome these two problems, an unconventional control structure to ensure propulsion system safety is proposed. The proposed solution is motivated by the difficulties encountered to measure the torque disturbances on propeller shaft eliminating the possibility of solving this problem by means of a conventional feedforward compensation strategy. Consequently, the association of a feedback master controller with a backstepping algorithm solves the problem satisfactory.

It is shown that it is possible to keep a high gain closed loop system stable with the contribution of a backstepping algorithm embedded into the loop. Additionally, the problem due to the contribution of the integral action of the feedback master controller (an undesired phase lag) is overcome by means of the backstepping algorithm that reacts accurately under the reference model response changes when the high gain proportional action of the feedback master controller detects disturbances.

Acknowledgements

The authors should like to acknowledge the contribution of the Spanish institution MICYT by the financial support to the grant DPI2003-00512.

References

Blanke, M., K. Lindegaard and T. I. Fossen (2000). Dynamic model for thrust generation of marine propellers. 5th IFAC Conference of Maneuvering and Control of Marine craft, MCMC pp. 363-368.

Blanke, M. Kinnaert, J. Lunze, M. Staroswiecki. (2003). Diagnosis and Fault-Tolerant Control. Springer-Verlag, Berlin. Heidelberg, pp 469-496

Fosen, T.I. and A. Grovlen (1998). Nonlinear Output Feedback Control of Dynamically Positioned Ships Using Vectorial Observer Backstepping. IEEE Transactions on Control Systems Technology, TCST-6(1):121-128, January 1998.

Fosen, T.I. and J.P. Strand (1999). Passive Nonlinear Observer Design for Ships Using Lyapunov Methods: Experimental Results with a Supply Vessel, Automatica. Aut-35(1), January 1999

Izadi-Zamanabadi R., and M. Blanke. (1998) A ship propulsion system as a benchmark for fault tolerant control. Technical report, Control Engineering dept.,Aalborg University, Denmark.

Izadi-Zamanabadi R., and M. Blanke. (1999), A ship propulsion system as a benchmark for fault tolerant control. Control Engineering Practice, 7(2):277-239.

Krstic M., I. Kanellakopoulos, and P. Kokotovic. (1995) Nonlinear and Adaptive Control Design. John Wiley and Sons, 1995.

Krstic M. and H. Deng (1998). Stabilization of Nonlinear Uncertain Systems (Springer Verlag, Berlin).

LaSalle, J.P (1966). Stability for Ordinary Differential Equations, Journal of Differential Equations, Vol. 4, pp. 57-65

MAN B&W Diesel A/S (2006), CEAS: Engine Room Dimensioning, 7S65ME-C Mk 8 with 2 x TCA66-20, SMCR: 20,090 kW at 95.0 RPM, Opt. point: 100.0 % , www.manbw.dk/ceas/erd

Minsaas, K.J., H.J. Thon and W. Kauczynski (1987).Estimation of Required Thruster Capacity for Operation of Offshore Vessels under Severe Weather Conditions, PRADS, Trondheim, Norway, pp. 411–427.

Pivano L. ,N. Smogeli, Thor Inge Fossen, Tor Arne Johansen (2006a), Experimental of a marine propeller thrust estimation scheme, MCM2006, Lisbon, Portugal

Pivano, L., T. I. Fossen and T. A. Johansen (2006b). Nonlinear model identification of a marine propeller over four-quadrant operations. 14th IFAC Symposium on System Identification, SYSID, Newcastle, Australia.

Smogeli, Ø. N., J. Hansen, A. J. Sørensen and T. A. Johansen (2004). Thrust optimal anti-spin control for marine propulsion systems. Proc. of the 43rd IEEE Conference on Decision and Control, Paradise Island, Bahamas

Smogeli, Ø. N., E. Ruth and A. J. Sørensen (2005). Experimental Validation of Power and Torque Thruster Control. 13th Mediterranean Conf. On Control and Automation (MED'05), Limassol, Cyprus

Sørensen, A.J., A.K. Ådnanes, T.I. Fossen and J.P. Strand (1997). A new method of thrusters control in positioning of ships based on power control. 4 th IFAC Conf. on Manoeuvring and Control of Marine Craft (MCMC'97), Brijuni, Croatia

Sontag E.D., A (, 1989),"universal" construction of Artestin's theorem on nonlinear stabilization. System and Control Letters, 13:117–123.

Yoshizawa, T. (1968). Stability Theory by Lyapunov's second Method, The Mathematical Society of Japan.

CHAPTER 4

The contribution of a nonlinear design to the identification process of the parameters that defined the ship´s dynamics

M. HARO CASADO[1] and R. FERREIRO GARCIA[2]

[1]Facultad de Ciencias Naúticas, Universidad de Cádiz, Polígono Rio San Pedro s/n, Edificio CASEM, 11510 Puerto Real. Cádiz
manuel.haro@uca.es
[2]Escuela Superior de la Marina Civil, Universidad de La Coruña, Paseo de Ronda 51,36011 La Coruña ferreiro@udc.es

The aim of this chapter is to show a reliable procedure to solve the problem of identification of the ship model parameters using the experimental data achieved from both manoeuvers, the change of course and the turning circle test. Two non-linear models (Bech and Norrbin) has been processed on the identification task to demonstrate the procedure suitability. Validations of the achieved results are based on the knowledge of the true parameter values, which were used to compare model based simulated results with the true observed data to show the validity of the proposed procedure.

1 Introduction

The Maritime Safety Committee (MSC) of the International Maritime Organization (IMO) has emitted in the year 1998 the Resolution MSC.74(69) whose title was “Adoption of New and Amended Performance Standards”. In its Annex 3 established the recommendations, general and specific, about the Universal Automatic Identification System (AIS). It is a shipboard transponder system in which ships continually transmit their ID, position, course, speed and other data to all other nearby ships and shoreside authorities on a common VHF radio channel. The concept is derived from the pioneering work of a Swedish inventor named Häkan Lans, who developed in the mid 1980s an ingenious technique for spontaneous, master-less communication, which permits a large number of transmitters to send data bursts over a single narrowband radio channel by synchronizing their data transmissions to a very precise timing standard.

AIS is designing to operate in one of the following modes:

- In a ship-to-ship mode for collision avoidance
- In a ship-to-shore mode as a means for coastal states to monitor and obtain information about a ship and its cargo
- As a traffic management tool when integrated with a Vessel Traffic System (VTS)

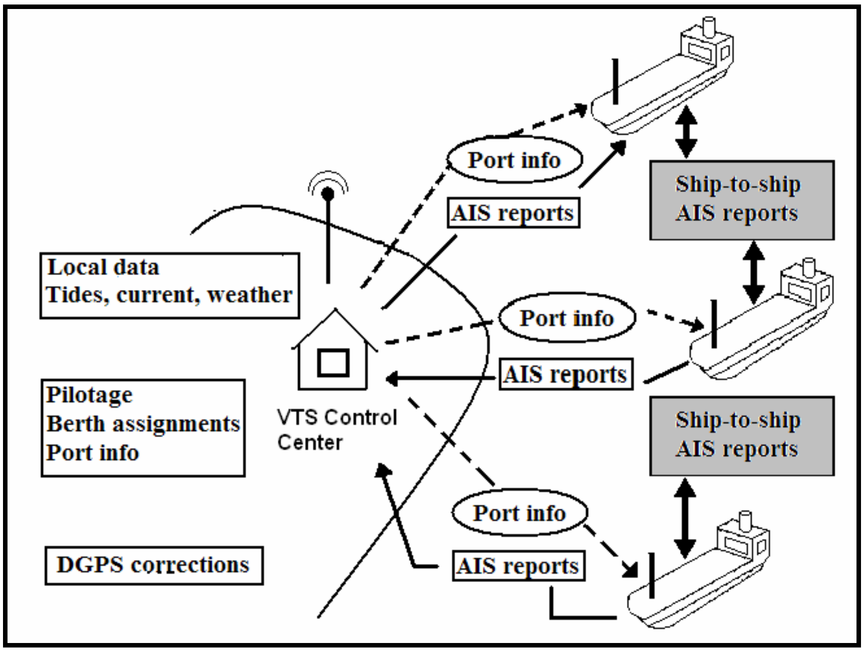


Fig. 1. AIS integrated with VTS (assumes control of timeslot assignments)

A new functionality can be added, the capacity of prediction about the foregone movements of the ship in the actual conditions of sailing under the external disturbances (wind, currents,..), state of load and under keel clearance. The resulting information can be transmitted by the AIS systems to other ships and the coastal stations. This objective can be carried out by an adequate identification algorithm of its dynamics that should work in real time. In this way the prediction of these movements can be determined and transmitted to other ships for collision avoidance as well as to the coastal stations. The identification algorithm utilised in this paper is based on the backstepping procedure design and a model that can represent in an adequate way the ship's dynamics as the non-linear models.

The new recursive design known as the adaptive backstepping (Krstić, et al., 1992) is based on three techniques which differ in the construction of adaptation law:

(i) Adaptive backstepping with overparametrization, when at each design step a new vector of adjustable parameters and the corresponding adaptation law are introduced (Kanellakopulos, et al.,1991),
(ii) Adaptive backstepping with modular identifiers, when a slight modification of the adaptive control allows independently the construction of estimation-based identifiers of unknown parameters (Krstić, et al., 1992),
(iii) Adaptive backstepping with tuning functions, when at each design step a virtual adaptation law known as tuning function is introduced, while the actual adaptation algorithm is defined at the final step in terms of all the previous tuning functions (Krstić, et al., 1995).

2 Ship Models

In the process of analysing the motion of a ship in 2 degrees of freedom (DOF) it is convenient to define two co-ordinate systems as indicated in Figure 2. The moving co-ordinate frame X_0 Y_0 is conveniently fixed to the ship and is denoted as the body-fixed frame. The origin of this body-fixed frame is usually chosen to coincide with the centre of gravity (CG) when CG is in the principal plane of symmetry. The earth-fixed co-ordinate frame is denoted as X Y. The angle Ψ is the course angle or heading, V_L is the forward velocity, V_T is the velocity in starboard direction and δ the rudder angle. The co-ordinates (x,y) denotes the ship's position along the track.

The elimination of the sway velocity (V_T) in the model of Davidson and Schiff (1946); Gerritsma (1980), has driven to the obtaining of the models of Nomoto (1957). Because of the assumption of mean constant forward speed in the models of Nomoto are only suitable for small rudders angles. Another phenomenon that cannot be described by these linear models is the effect of course instability. For these reasons, later on Norrbin (1970) and Bech (1969), substituted the linear term of the angular acceleration of the ship in the Nomoto's first and second order, respectively, for a non-linear term formed by a polynomial of third order whose coefficients were determined by the Bech's reverse spiral manoeuvre.

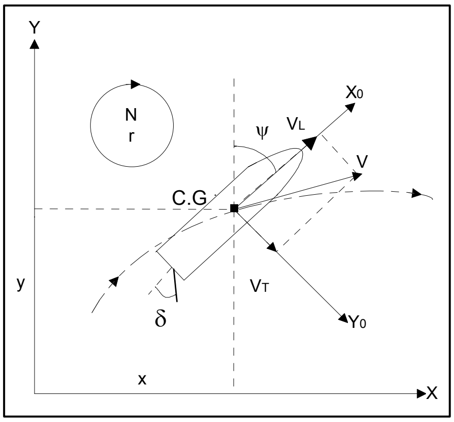


Fig. 2. Variables used in the description of the ship's movement

In basis on the cinematic variables define in Fig. 2, the motion equations that can to define the ship movements are

$$\dot{x} = V_L \cdot \sin\psi + V_T \cdot \cos\psi \tag{1.a}$$

$$\dot{y} = V_L \cdot \cos\psi - V_T \cdot \sin\psi \tag{1.b}$$

Where the speed rate in the longitudinal direction or surge speed and the transversal or sway one, V_L and V_T, respectively are

$$\dot{V}_L = -d \cdot V_L - e \cdot V_\psi^2 + S \tag{2.a}$$

$$V_T = -f \cdot V_\psi - g \cdot V_\psi^3 \tag{2.b}$$

being V_ψ the ship's speed in the angular sense. The equation (2.a) indicates how the propeller's thrust resulting in a forward acceleration S is counteracted by the ship speed as well as the turning rate. The relation between the speed V_T and the rate of turn V_ψ, determined and affirmed by the full scale and laboratory experiments is given by the equation (2.b).

The first identification algorithm utilised in this chapter is based on the backstepping procedure and in a model that can represent in an adequate way the ship’s dynamics as the model of Bech. It takes the form

$$T_1 \cdot T_2 \cdot \dddot{\psi} + (T_1 + T_2) \cdot \ddot{\psi} + K \cdot H_B(\dot{\psi}) = K \cdot U \qquad (3.a)$$

$$H_B(\dot{\psi}) = b_0 + b_1 \cdot \dot{\psi} + b_2 \cdot \dot{\psi}^2 + b_3 \cdot \dot{\psi}^3 \qquad (3.b)$$

where ψ is the ship course, $r = \dot{\psi}$, the rate of turn and $U = \delta + T_3 \cdot \dot{\delta}$ is the control, being δ the rudder angle.

The equation (3.a) is an extension of the Nomoto’s second order model that has been amplied to include non-linear effects by adding dynamic non-linearities referred to as manoeuvring characteristics gived by a non-linear function $H_B(r)$ (3.b).

In order to carry out the system description by the nonlinear state equations it is preferable to define the following state variables. $x_1 = \dot{\psi} = r$ (yaw rate), $x_2 = \ddot{\psi} = \dot{r}$ (angular acceleration), being the output y. The dynamic equations of ship dynamics are,

$$\dot{x}_1 = x_2 \qquad (4.a)$$

$$\dot{x}_2 = d \cdot U + \varphi^T \cdot \theta \qquad (4.b)$$

$$y = x_1 \qquad (4.c)$$

being the coefficient’s values,

$$a = -\frac{T_1 + T_2}{T_1 \cdot T_2} \qquad (5.a)$$

$$c_1 = -\frac{K \cdot b_1}{T_1 \cdot T_2} \qquad (5.b)$$

$$c_2 = -\frac{K \cdot b_2}{T_1 \cdot T_2} \qquad (5.c)$$

$$c_3 = -\frac{K \cdot b_3}{T_1 \cdot T_2} \qquad (5.d)$$

$$c_0 = -\frac{K \cdot b_0}{T_1 \cdot T_2} \qquad (5.e)$$

$$d = \frac{K}{T_1 \cdot T_2} \tag{5.f}$$

The known nonlinearitie φ and the vector θ of constant and unknown parameters are,

$$\varphi^T = \begin{bmatrix} 1 & x_2 & x_2^2 & x_2^3 & x_3 \end{bmatrix} \tag{6.a}$$

$$\theta^T = \begin{bmatrix} c_0 & c_1 & c_2 & c_3 & a \end{bmatrix} \tag{6.b}$$

The block diagram of the system tested is showed in Fig. 3. Systems of this form are called parametric strict feedback systems or triangular systems (by analogy with linear systems), because the nonlinearities depend only on the variables and parameters that are feed back.

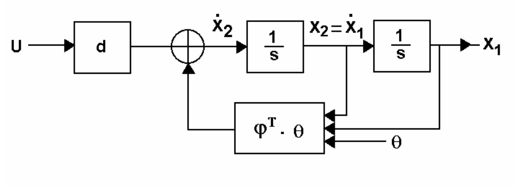


Fig. 3. Block diagram of a second-order analysed system

Finally, a rudder control with first-order actuator dynamics is modelled as

$$\dot{\delta} = -\frac{1}{T_3} \cdot (\delta - U) \tag{7}$$

The second identification algorithm that has been developed is based on both the Norrbin model and the turning circle test. In this case the ship steering dynamics can be which is an extension of the first-order model of Nomoto that has been amplied to include non-linear effects by means of the consideration of a function that describes the non-linear nature of ship dynamics and can be expressed in terms of a polynomial expansion as $H_N(r) = -a_0 - a_1 \cdot r - a_2 \cdot r^2 - a_3 \cdot r^3$.

$$\dot{\psi} = r \tag{8.a}$$

$$\dot{r} = -a_0 - a_1 \cdot r - a_2 \cdot r^2 - a_3 \cdot r^3 \tag{8.b}$$

3 Ship Under Test

The ship whose dynamic is determined by the procedures summarized in this chapter has been built in the shipyard of Navantia (2001) localized in the Cádiz bay in the year 2001.

Table 1. Main characteristics of the ship tested

Name	Stena Holландica	Bulbous bow	Yes
Draught forward (full load condition)	6 m	Type of rudder	Becker (2 units)
Draught aft (full load condition)	6 m	Maximum angle of rudder	65 degrees
Deadweight	7456 metric tonnes	Time of hard–over to hard over	56 s
Maximum displacement	19949 metric tonnes	Propellers	2
Length overall	188.3 m	Engine (2 per shaft) at maximum power	4 x 6000 kW
Breadth (moulded)	28.7 m	Speed loaded (maximum full ahead)	23.14 knots

The experimental results obtained from the manoeuvring documentation (according the IMO Resolution A.601(15) - interim standards for ship manoeuvrability) with the ship are in the normal ballast condition, with a surge speed of 23.14 knots and a rudder angle of 20 degrees are shown in Table 2.

Table 2. Course change test results

Time from wheel over position	Change of heading		Speed after turn
(s)	(deg)	(rad)	(knots)
0	0	0	23.14
15.03	10	0.1745	19.46
22.07	20	0.3491	18.84
28.69	30	0.5236	18.14
35.43	40	0.6981	17.33
42.29	50	0.8727	16.55
49.36	60	1.0472	15.74
56.64	70	1.2217	14.95
64.25	80	1.3963	14.17
72.00	90	1.5708	13.44
80.00	100	1.7453	12.74
88.00	110	1.9199	12.12
96.50	120	2.0944	11.51

In the tested ship the rudder (2 units of Becker type) can be moved from 65 degrees port to 65 degrees starboard within 56 seconds, thus for a step input in the rudder angle of 20 degrees it is necessary a time of 8.3 s to reach the 99.3% of the steady state for the rudder angle δ according to the linear dynamics of first order represented by the equation (7). In other words the constant time is T_3=1.7 s.

4 Initial Identification Processes

Two design procedures has been developed to carry out the identification of the parameters in the non-linear model. The first is based on recording the temporal variation of the yaw angle during a simple maneuver of change; the second in the determination of the radius angle when the trial of the turning test is carried out.

To establish the goodness of the proposed procedure of identification it is necessary to know the true values of the parameters (supposed constants) that appear in the dynamical equations of Bech and Norrbin that has been utilised in this work. For this purpose the goal to be achieved is to reduce the difference between the experimental values obtained from the experimental course change speed (showed in Table 2) and the solutions of the non-linear differential equations (3.a) and (8.b) by means of an appropriate selection of the equation's parameters. The procedure is based on the following procedures and algorithms:

- In basis to the values that are shown in Table 2 the dependency of the yaw angle and the yaw rate on the time has been obtained by means of fitting by least square procedure of the experimental points to a polynomial of order six. The temporal variation of the yaw angle was obtained by differentiation of this result.

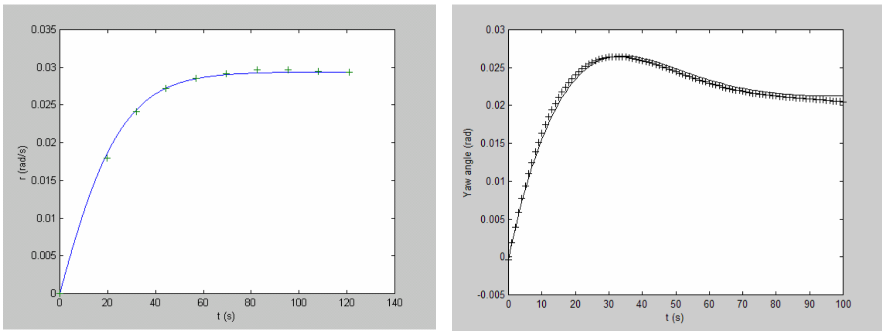
$$\psi_d = 0.2191 \cdot 10^{-11} \cdot t^6 - 0.1125 \cdot 10^{-8} \cdot t^5 + 0.2278 \cdot 10^{-6} \cdot t^4 - 0.2287 \cdot 10^{-4} \cdot t^3 + 0.1135 \cdot 10^{-2} \cdot t^2 - 0.3382 \cdot 10^{-3} \cdot t - 0.1519 \cdot 10^{-2} \tag{9.a}$$

$$r_d = y_d(t) = 0.1315 \cdot 10^{-11} \cdot t^5 - 0.5627 \cdot 10^{-8} \cdot t^4 + 0.9112 \cdot 10^{-6} \cdot t^3 - 0.6862 \cdot 10^{-4} \cdot t^2 - 0.2270 \cdot 10^{-2} \cdot t - 0.3382 \cdot 10^{-2} \tag{9.b}$$

where r_d is the rate of turn in (rad/s) and Ψ the course (rad).

- A Backward-Euler integration algorithm with a step size of 1 second.
- An optimisation algorithm of Powell (Darnell, 1990) with an optimisation criteria ITAE (assuming the ITAE performance criterion as the product of integral of time-weighted absolute error).
- A trial and error procedure.

The resulting parameters are shown in Table 3, while in Fig. 4, the experimental yaw angle and yaw rate is compared with the theoretical ones, showing an excellent agreement between them. The purpose of this paper is to design a systematic procedure for finding an identification algorithm that does not depend, even partially, on a heuristic method such as the one used to know the true values of the ship's model parameters and thus it is possible to know the excellence of the proposed procedure. The procedure of the identification based on the tuning functions design resembles the classical procedure used in the model reference adaptive control (MRAC) (both are based in the Liapunov stability theory) but, however, it shows a considerable advantage over the traditional scheme. Both for a same control effort and initial conditions, the transient performance of tuning functions is far superior, whereas the tracking error with the tuning functions scheme is only a fraction of the indirect method. This is a consequence of incorporating the parameter update law into the controller (Kanellakopoulos, et al., 1991). Other two most important factors which contribute to the superior performance of the tuning functions scheme are nonlinear damping and reference model initialisation.



(a) (b)

Fig. 4. The experimental values (+) of the yaw rate (a) and the yaw angle (b) are compared with the fittings (continuous line)

The procedure described in this paper can be used as a new and alternative
estimation method on the traditional ones, Continuous Least-Squares, Recursive Least- Squares , Recursive Maximum Likelihood, State Augmented Kalman Filter (Fossen, 1994).

5 Identification Based on the Change of Yaw Angle

The identification objective consist in the determination of the parameters of the dynamic equation of motion (T_1, T_2, K) and the coefficients of the nonlinear manoeuvring equation $\left[c_i\ (i = 0..3), a\right]$ supposed constants during the period of recollection of data ≈ 100 s. The general-purpose computer system measures data of the ship sensor (based on the GPSD) and in basis on the algorithms that will be developed determines these parameters. The information is transmitted to the autopilot for the control of the steering. If the environmental conditions of sailing change during the following period the new dynamics is updated and an indirect adaptive steering is carried out. The identification procedure is based on the backstepping procedure developed by Krstić in 1995.

In the backstepping approach, two additional variables are introduced. The variable z_1 representing the tracking error, and z_2 which means the error variable that expresses the fact by which x_2 is not the true control, both are defined by:

$$z_1 = x_1 - x_d \tag{10.a}$$

$$z_2 = x_2 - \beta(z_1) \tag{10.b}$$

where the function $\beta(z_1)$ represents a stabilizing function whose dependence will be chosen later with the objective of achieving the asymptotic stability.

In the first step of the backstepping it is necessary to derive the equation (10.a) and with the aid of the (4.a) and (10.b) to obtain

$$\dot{z}_1 = z_2 + \beta(z_1) - \dot{y}_d \tag{11}$$

and choosing the stabilizing function as

Table 3.

	Bech's model			Norrbin's model	
Parameter	Value	Unit	Parameter	Value	Unit
K	$6.4817 \cdot 10^{-2}$	s	–	–	–
T_1	112.023	s	–	–	–
T_2	13.5707	s	–	–	–
T_3	1.70	s	–	–	–
c_0	$-7.77 \cdot 10^{-7}$	$\text{rad} \cdot \text{s}$	a_0	$3.10284 \cdot 10^{-3}$	$\text{rad} \cdot \text{s}^{-1}$
c_1	$2.39669 \cdot 10^{-3}$	s^{-2}	a_1	$1.00982 \cdot 10^{-2}$	s^{-1}
c_2	-0.146117	$\text{rad}^{-1} \cdot \text{s}^{-1}$	a_2	1	rad^{-1}
c_3	$2.44894 \cdot 10^{-5}$	rad^{-2}	a_3	0.77	$\text{s} \cdot \text{rad}^{-2}$
a	$-8.26149 \cdot 10^{-2}$	s^{-1}	–	–	–
d	$4.27019 \cdot 10^{-5}$	s	–	–	–

$$\beta(z_1) = \dot{y}_d - K_1 \cdot z_1 - \beta_1(z_1) \cdot z_1 \tag{12}$$

being $K_1 > 0$, $\beta_1(z_1) \geq 0, \forall z_1$. In this step a Liapunov's function candidate is,

$$V_1 = \frac{1}{2} \cdot z_1^2 \tag{13}$$

whose derivative is

$$\dot{V}_1 = -[K_1 + \beta_1(z_1)] \cdot z_1^2 + z_1 \cdot z_2 \tag{14}$$

The second step of the backstepping depart of the equation (10.b) where the variation of the second backstepping state is,

$$\dot{z}_2 = \dot{\alpha} - \dot{\beta}(z_1) = \sum_{i=0}^{3} c_i^i + a \cdot x_2 + d \cdot U - \dot{\beta}(z_1) \tag{15}$$

As consequence of the uncertainties in the parameters c_i (i=0..3) and a, it cannot be cancelled by the control. For the principle of certain equivalence, each one of the parameters are substituted for its estimate $\hat{c}_i$ (i=1..3) and $\hat{a}$, the made errors are

$$\tilde{c}_i = c_i - \hat{c}_i \qquad (i = 0..3) \tag{16.a}$$

$$\tilde{a} = a - \hat{a} \tag{16.b}$$

The second Liapunov function proposed is,

$$V_2 = V_1 + \frac{1}{2} \cdot z_2^2 + \frac{1}{2} \cdot \left[\sum_{i=0}^{3} \frac{1}{\gamma_i} \cdot \tilde{c}_i^2 \right] + \frac{1}{2} \cdot \frac{1}{\gamma_a} \cdot \tilde{a}^2 \tag{17}$$

its derivative is

$$\dot{V}_2 = \dot{V}_1 + z_2 \cdot \dot{z}_2 + \sum_{i=0}^{3} \frac{1}{\gamma_i} \cdot \tilde{c}_i \cdot \dot{\tilde{c}}_i + \frac{1}{\gamma_a} \cdot \tilde{a} \cdot \dot{\tilde{a}} \tag{18}$$

after of considering (14,15,16.a.b) the equation (18) can be transformed in

$$\dot{V}_2 = -\left[K_1 + \beta_1(z_1)\right]\cdot z_1^2 + z_1\cdot z_2 + z_2\cdot\left[\hat{a}\cdot x_2 + \sum_{i=0}^{3}\hat{c}_i\cdot x_i^i + d\cdot U - \dot{\beta}\right] +$$
$$\left[z_2\cdot x_2 + \frac{1}{\gamma_a}\cdot\dot{\tilde{a}}\right]\cdot a + \sum_{i=0}^{3}\left(z_2\cdot x_i^i + \frac{1}{\gamma_i}\cdot\dot{\tilde{c}}_i\right)\cdot\tilde{c}_i \tag{19}$$

in a real situation it is predictable that $\sum_{i=0}^{3} x_i^i \neq 0$, that it to say, always there is an error in the parameter estimations, in consequence to eliminate the last term in (19), two solutions can be adopted,

- solution less restrictive

$$\sum_{i=0}^{3} x_i^i\cdot z_2 + \sum_{i=0}^{3}\frac{1}{\gamma_i}\cdot\dot{\tilde{c}}_i = 0 \tag{20.a}$$

- solution more restrictive,

$$x_i^i\cdot z_2 = -\frac{1}{\gamma_i}\cdot\dot{\tilde{c}}_i \qquad (i = 0\cdots 3) \tag{20.b}$$

If someone of the conditions (20.a) or (20.b) are match, then of the (19)

$$\dot{V}_2 = -\left[K_1 + \beta_1(z_1)\right]\cdot z_1^2 + z_1\cdot z_2 + z_2\cdot\left[\hat{a}\cdot x_2 + \sum_{i=0}^{3}\hat{c}_i^i\cdot x_i^i + d\cdot U - \dot{\beta}\right] \tag{21}$$

If the rudder angle is chosen as,

$$U = d^{-1}\cdot\left\{-z_1 - \hat{a}\cdot x_2 - \sum_{i=0}^{3}\hat{c}_i\cdot x_i^i + \dot{\beta} - \left[K_2 + \beta_2(z_2)\right]\cdot z_2\right\} \tag{22}$$

the derivative of the Liapunov function (19) is now

$$\dot{V}_2 = -[K_1 + \beta_1(z_1)] \cdot z_1^2 - [K_2 + \beta_2(z_2)] \cdot z_2^2 \tag{23}$$

if the functions β_1 and β_2 are chosen as

$$\beta_1(z_1) = z_1^2 \tag{24.a}$$

$$\beta_2(z_2) = z_2^2 \tag{24.b}$$

and $K_1 > 0, K_2 > 0$. Since $\dot{V}_2$ is negative definite, it follows from LaSalle (1968)-Yoshizawa (1966) theorem, that in the (z_1, z_2) coordinates the equilibrium (0,0) is global asymptotically stable (GAS) and $\tilde{a}, \tilde{c}_i$ $(i = 0 \cdots 3)$ are global uniformly bounded. In view of (10.a), $x_1(t)$ tends to $y_d(t)$ asymptotically. From (10.b) the condition $z_2 \to 0$ as $t \to \infty$ implies that $x_2 \to \beta(z_1)$.

As $x_2 = \dot{x}_1 = \dot{y}$ and, as $\beta(z_1) = \dot{y}_d - K_1 \cdot z_1 - \beta(z_1) \cdot z_1, z_1 \to 0$, $\dot{y} \to \dot{y}_d$. That is to say, the temporal variation of the actual trajectory tends to desired one which is supposed to be bounded in an asymptotically manner.

The dynamics of the first error state can be obtained from (11) and considering (12). The result is

$$\dot{z}_1 = -[K_1 + \beta_1(z_1)] \cdot z_1 + z_2 \tag{25}$$

the corresponding variation of the state z_2 , is obtained to depart of (15) with the control chosen (22)

$$\dot{z}_2 = -z_1 - [K_2 + \beta_2(z_2)] \cdot z_2 + +\tilde{a} \cdot x_2 + \sum_{i=0}^{3} \tilde{c}_i \cdot x_i^i \tag{26}$$

in matrix form after of taking into account (25) and (26),

$$\begin{bmatrix} \dot{z}_1 \\ \dot{z}_2 \end{bmatrix} = \begin{bmatrix} -[K_1 + \beta_1(z_1)] & 1 \\ -1 & -[K_2 + \beta_2(z_2)] \end{bmatrix} \begin{bmatrix} z_1 \\ z_2 \end{bmatrix} + \begin{bmatrix} 0 \\ \tilde{a} \cdot x_2 + \sum_{i=0}^{3} \tilde{c}_i \cdot x_i^i \end{bmatrix} \tag{27}$$

For the implementation of the control signal (22) it is necessary to express the functions β and its derivative in terms of the variables previously defined. It is easy to show that these are defined by the equations,

$$\beta = \dot{y}_d - K_1 \cdot z_1 - z_1^3 \tag{28.a}$$

$$\dot{\beta} = \ddot{y}_d - \left[K_1 + \frac{\partial \left[\beta_1(z_1) \cdot z_1 \right]}{\partial z_1} \right] \cdot \dot{z}_1 \tag{28.b}$$

After using the damping functions (24.a) and (24.b),

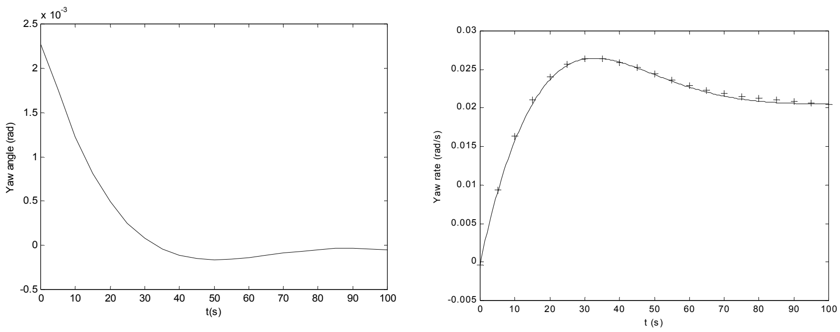
$$\dot{\beta} = \ddot{y}_d - \left[K_1 + 3 \cdot z_1^2 \right] \cdot \dot{z}_1 = \ddot{y}_d - \left[K_1 + 3 \cdot z_1^2 \right] \cdot \left[-\left(K_1 + z_1^2 \right) \cdot z_1 + z_2 \right] \tag{29}$$

5.1 Identification Results

The simulation of the system has been carried out. By means of the algorithm of integration of Backward- Euler with a step-size of 0.1 s. The optimisation one that allows the reduction of the error between the parameters estimates and its true values was of the Fletcher Reeves, that requires few iterations to convergence (Flannery, *et al*., 1989). The procedure is capable of identifying the parameters that are presents in the dynamic equation of the ship movement (3.a.b). The results are shown in Table 4 jointly with the values of the gains utilised in the identification process.
One of the simulation result is shown in Fig. 5.b where the experimental values (symbol +) are compared with the variation of the yaw rate obtained by the summarised procedure showed in the paper (continuous line). The initial estimations of the parameters were the same for all model parameters (20% of its true values). The maximum error committed in the estimation process was of 0.9%. In Figs.6.a-b,7 appears the fast convergence of the estimation errors toward the null values.

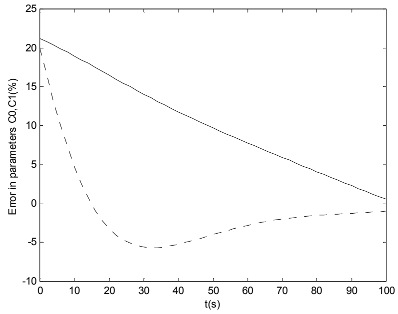
Table 4.

Parameter	Value	Units	Controller gains (in p.u)	Value
–	–	–	K_1	32.09
c_0	$-7.77 \cdot 10^{-7}$	$rad \cdot s$	K_2	71.06
c_1	$2.5 \cdot 10^{-3}$	s^{-2}	γ_0	$1.50 \cdot 10^{-7}$
c_2	-0.146117	$rad^{-1} \cdot s^{-1}$	γ_1	-1.93
c_3	$-2.44894 \cdot 10^{-2}$	rad^{-2}	γ_2	1250
a	$-36149 \cdot 10^{-2}$	s^{-1}	γ_3	$-1.1 \cdot 10^{5}$
d	$4.27873 \cdot 10^{-5}$	s	γ_a	58.78

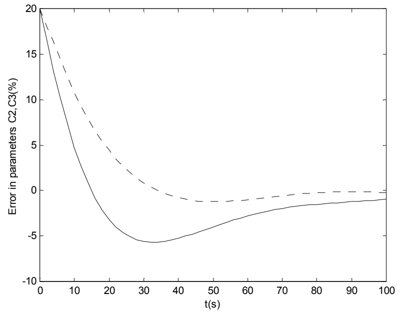


(a) (b)

Fig. 5.(a) Temporal variation of the yaw angle. (b) Comparison between the experimental values of the yaw rate (+) and the simulation result (continuous line)



(a)



(b)

Fig. 6. (a)Variations of the errors on the parameters [c_0(-),c_1(- -)].(b) Variations of the errors on the parameters [c_2(-),c_3(- -)].

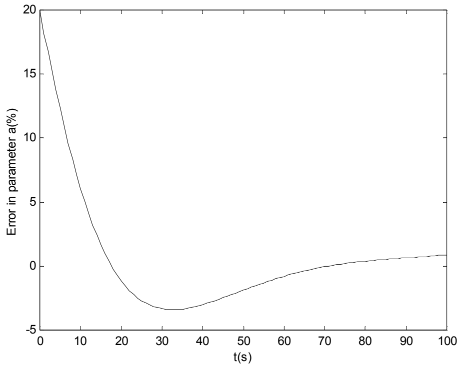


Fig. 7. Variations of the errors on the parameter (a)

6 Identification Based on the Turning Circle Test

Several trial test for marine vehicles has been proposed, namely, the Turning, the Z-Manoeuvre (Kempf), Modified Z-Maneuver, Direct Spiral (Dieudonné), Reverse Spiral (Bech), Pull-Out, Stopping, Stopping Inertia, New Course Keeping, Man-Overboard, Parallel Course Manoeuver, Inertial Turning, Z-Maneuver Test at Low Speed, Accelerating Turning, Acceleration/Deceleration, Thruster, Minimum Revolution, Crash Ahead Test, though there are no definitive international standards for conducting manoeuvring trials with ships. Many shipyards have developed their own procedures driven by their experience and with consideration to the efforts made by the International Towing Tank Conference (ITTC, Proceedings 1963-1975) and other organizations or institutes (Journée and Pinkster, 2001). The society of Naval Architects and Marine Engineers (SNAME) has produced three guidelines: “Code on manoeuvring and Special Trials and Tests” (1950), “Code for Sea Trials” (1973) and “Guide for Sea Trials” (1989). The Norwegian Standard Organization has produced “Testing of New Ship, Norsk Standard” (1985). The Japan Ship Research Association (JSRA) has produced a “Sea Trial Code for Giant Ships (1972). IMO Resolution A.601 (1987) and IMO Resolution A.751 (1993), were adopted by the IMO Assemblies to address ship manoeuvrability. The last Resolution adopted by this organization was the MSC 137(76) on 4 December 2002. The Z - Maneuver Test (Kemf) jointly with the Spiral Manoeuvre give some indication of control effectiveness (yaw- angle rate versus rudder angle) are recommended for all organisations and let us to carry out a compare the manoeuvring properties and control characteristic of a ship with those of other ships.

Experimental results of trial tests can be to meet in the references showed in Principles of Manoeuvring and Control (Crane,1999). In tested ship the results of the turning circle test are showed in Table 5 .

The backstepping design is capable of realising this task of identifying The identification procedure only needs the value of the radius in the turning test trials. The procedure can be to extend to the determination of the term c that multiplies to the rudder angle in the Norrbin equation that describes the ship motion.

Table 5. Values obtained under the turning circle test. (Max. ahead speed 22.68 knots, normal ballast condition, 20 degrees of rudder). (For details see the Figure 8)

t (s)	Transfer (m)	Advance (m)
0	0	0
19.67	0	203.72
32.00	18.52	314.84
44.38	55.56	444.48
56.88	111.12	555.60
69.67	166.68	648.20
82.38	259.28	722.28
95.25	351.88	796.38
108.00	463.00	851.92
121.00	574.12	886.96

6.1 Description of the Circular Motion

It should be considered that the transversal speed can't considered zero during a circular movement because a non-zero transverse velocity is a prerequisite for turning the ship along a circular path, such as the turning circle, however seemed reasonable to suppose that the condition should be stayed during the entire movement $V=V_L>>V_T$. Below this assumption the equations (1.a-b) are

$$\dot{x} \approx V_L \cdot \sin\psi \tag{30.a}$$

$$\dot{y} \approx V_L \cdot \cos\psi \tag{30.b}$$

In a radial movement the cinematic equations are (see for details the Fig.8),

$$x = x_0 + R \cdot \sin\gamma \tag{31.a}$$

$$y = y_0 + R \cdot \cos\gamma \tag{31.b}$$

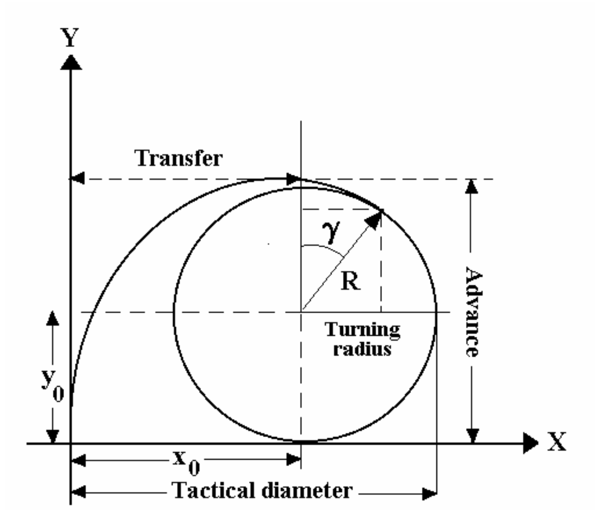


Fig. 8. Main parameters that defined the turning circle test

resolving the equations (30-31.a-b) in terms of $\dot{R}$ and $\dot{\gamma}$ it is easy to probe that the temporal variations of the

$$\dot{\gamma} = \frac{V_L}{R} \cdot \sin(\psi - \gamma) \tag{32.a}$$

$$\dot{R} = V_L \cdot \cos(\psi - \gamma) \tag{32.b}$$

6.2 Statement of the Problem under Space State Description

For this purpose the following state variables are introduced: $x_1 = R - R_d = \tilde{R}$, where R_d is the radius that is described by the ship during the turning circle and R the actual one while $\tilde{R}$ represents the error between these two variables; $x_2 = \psi$ (yaw angle), $x_3 = r$ (yaw rate), $x_4 = \gamma$ (angular position in the circular motion).With those choices the state equations are,

$$\dot{x}_1 = V \cdot \cos(x_2 - x_4) \tag{33.a}$$

$$\dot{x}_2 = x_3 \tag{33.b}$$

$$\dot{x}_3 = -\sum_{i=0}^{3} a_i \cdot x_3^i + c \cdot \delta \tag{33.c}$$

$$\dot{x}_4 = \frac{V}{x_1 + R_d} \cdot \sin\left(x_2 - x_4\right) \tag{33.d}$$

6.3 Adapative Backstepping Procedure and Identification One

Initially on the recursive procedure, the state x_2 is treated as a virtual control for the equation (33.a). At each subsequent step, it will be increased the designed subsystem by one equation. At the i- step, the ith- order subsystem is stabilised with respect to a Liapunov function V_i by the design of a stabilising function α_i. The updating law of the adaptive control system that allows us to know the true values of the dynamic model and the control signal is designed at the final step. To implement the identification procedure based on backstepping , following steps are to be performed:

STEP 1

The following variables are introduced:

$$z_1 = x_1 \tag{34.a}$$

$$z_2 = V \cdot \cos\left(x_2 - x_4\right) - \alpha_1 \tag{34.b}$$

where α_1 is using as a control to stabilize the z_1 subsystem, we choose the following Liapunov function candidate,

$$V_1 = \frac{1}{2} \cdot z_1^2 \tag{35}$$

its derivative is given by

$$\dot{V}_1 = z_1 \cdot \left(z_2 + \alpha_1\right) \tag{36}$$

choosing the stabilising function α_1 as a simple linear feedback law

$$\alpha_1 = -K_1 \cdot z_1 = -K_1 \cdot x_1 = -K_1 \cdot \left(R - R_d\right) \tag{37}$$

which leads to

$$\dot{V}_1 = -K_1 \cdot z_1^2 + z_1 \cdot z_2 \tag{38}$$

the second term $z_1.z_2$ in (38) will be cancelled at the next step.

STEP 2

It is necessary to consider that state x_3 is the control variable in the second equation (33.b). The third backstepping variable z_3 is defined by the equation,

$$z_3 = x_3 - \alpha_2 \tag{39}$$

where α_2 is a second stabilizing function used as control to stabilize the (z_1,z_2) subsystem. In this step is possible to meet the variation of the second error variable z_2, without more than to take its derivative in (34.b), after of considering (33.b) and (33.d), to meet

$$\dot{z}_2 = -V \cdot \left[x_3 - \frac{V}{x_1 + R_d} \cdot \sin\left(x_2 - x_4\right) \right] \cdot \sin\left(x_2 - x_4\right) - \dot{\alpha}_1 \tag{40}$$

It is important to observe that the time derivative $\dot{\alpha}_1$ can be implemented analytically without a differentiator in the following manner,

$$\dot{\alpha}_1 = \frac{\partial \alpha_1}{\partial t} = \frac{\partial \alpha_1}{\partial R} \cdot \frac{\partial R}{\partial t} = -K_1 \cdot \dot{R} \tag{41}$$

In this step the initial Liapunov function is augmented with a quadratic term

$$V_2 = V_1 + \frac{1}{2} \cdot z_2^2 \tag{42}$$

its temporal variation after of considering the equations (38) and (40),

$$\dot{V}_2 = \left(-K_1 \cdot z_1^2 + z_1 \cdot z_2\right) - z_2 \cdot z_3 \cdot V \cdot sin\left(x_2 - x_4\right) - z_2 \cdot \alpha_2 \cdot sin\left(x_2 - x_4\right) + \frac{z_2 \cdot V^2}{x_1 + R_d} \cdot sen^2\left(x_2 - x_4\right) - z_2 \cdot \dot{\alpha}_1 \tag{43}$$

If $\dot{V}_2$ must be negative definite in terms of z_1 and z_2, the second stabilising function α_2 is chosen as,

$$\alpha_2 = \frac{1}{V \cdot \sin\left(x_2 - x_4\right)} \cdot \left[z_1 + K_2 \cdot z_2 + \frac{V^2 \cdot \sin^2\left(x_2 - x_4\right)}{x_1 + R_d} - \dot{\alpha}_1 \right] \tag{44}$$

in this way

$$\dot{V}_2 = -K_1 \cdot z_1^2 - K_2 \cdot z_2^2 - z_2 \cdot z_3 \cdot V \cdot \sin\left(x_2 - x_4\right) \tag{45}$$

STEP 3

In this last step we can

$$\dot{z}_3 = \dot{x}_3 - \dot{\alpha}_2 = -\sum_{i=0}^{3} a_i \cdot x_3^i + c \cdot \delta - \dot{\alpha}_2 \tag{46}$$

The last Liapunov function utilised is

$$V_3 = V_2 + \frac{1}{2} \cdot z_3^2 + \frac{1}{2} \cdot \left[\sum_{i=0}^{3} \frac{1}{\gamma_i} \cdot \tilde{a}_i^2 \right] \tag{47}$$

where $\tilde{a}_i \left(i = 0 \cdots 3\right)$ represents the estimation errors, differences between the true values $a_i \left(i = 0 \cdots 3\right)$ and the estimation ones $\hat{a}_i \left(i = 0 \cdots 3\right)$.

$$\dot{V}_3 = -K_1 \cdot z_1^2 - K_2 \cdot z_2^2 - z_2 \cdot z_3 \cdot V \cdot sin\left(x_2 - x_4\right) + z_3 \left[-\sum_{i=0}^{3} \hat{a}_i \cdot x_3^i + c \cdot \delta - \dot{\alpha}_2 \right] + \sum_{i=0}^{3} \frac{1}{\gamma_i} \cdot \tilde{a}_i \cdot \dot{\tilde{a}}_i - \tilde{a}_i \cdot x_3^i \cdot z_3 \tag{48}$$

Using the control δ and an updating law to stabilize the entire system (z_1,z_2,z_3). We choose the control as,

$$\delta = \frac{1}{c} \cdot \left[-K_3 \cdot z_3 + z_2 \cdot V \cdot \sin\left(x_2 - x_4\right) + \sum_{i=0}^{3} \hat{a}_i \cdot x_3^i + \dot{\alpha}_2 \right] \tag{49}$$

$$\dot{V}_3 = -\sum_{i=1}^{3} K_i \cdot z_i^2 + \sum_{i=0}^{3} \tilde{a}_i \cdot \left[\frac{1}{\gamma_i} \cdot \dot{\tilde{a}}_i - x_3^i \cdot z_3 \right] \tag{50}$$

The terms that contains the parameter error are now eliminated with the update laws

$$\frac{1}{\gamma_i} \cdot \dot{\tilde{a}}_i = x_3^i \cdot z_3 \qquad \left(i = 0 \cdots 3\right) \tag{51}$$

This equation can be developed in the ones,

$$i = 0 \qquad \dot{\tilde{a}}_0 = \gamma_0 \cdot z_3 \tag{52.a}$$

$$i = 1 \qquad \dot{\tilde{a}}_2 = \gamma_1 \cdot x_3 \cdot z_3 = \gamma_1 \cdot r \cdot z_3 \tag{52.b}$$

$$i = 2 \qquad \dot{\tilde{a}}_2 = \gamma_2 \cdot x_3^2 \cdot z_3 = \gamma_2 \cdot r^2 \cdot z_3 \tag{52.c}$$

$$i = 3 \qquad \dot{\tilde{a}}_3 = \gamma_3 \cdot x_3^2 \cdot z_3 = \gamma_3 \cdot r^3 \cdot z_3 \tag{52.d}$$

Previous described laws guarantee that

$$\dot{V}_3 = -\sum_{i=1}^{3} K_i \cdot z_i^2 \le 0 \tag{53}$$

Since $\dot{V}_3$ is negative definite, it follows from LaSalle-Yoshizawa theorem (LaSalle,1968; Yoshizawa, 1966), that in the (z1,z2,z3) coordinates the equilibrium (0,0,0) is global asymptotically stable, and $\hat{a}_i(t), i = 0 \cdots 3$, are global uniformly bounded. In view of (34.a), $z_l = x_l$ go to zero asymptotically and $R \to R_d$ as $t \to \infty$. From (34.b) for a V bounded and $\alpha_1 = -K_1 \cdot z_1$ this implies that also x_2 towards zero an asymptotically manner. From (39), x_3 tends towards α_2 (the second stabilizing function).

6.4 Error Dynamics

For implementation purposes is convenient to obtain the error dynamics in the states previously introduced. The first dynamic is obtained to depart of (34.a-b) and (37),

$$\dot{z}_1 = \dot{x}_1 = V \cdot \cos(x_2 - x_4) = z_2 + \alpha_1 = -K_1 \cdot z_1 + z_2 \tag{54}$$

The second equation can be obtained after of considering the equations (40) and (44),

$$\dot{z}_2 = -z_1 - K_2 \cdot z_2 - V \cdot z_3 \cdot \sin(x_2 - x_4) \tag{55}$$

The third equation is obtained (46) and (49)

$$\dot{z}_3 = -\sum_{i=0}^{3} \tilde{a}_i \cdot x_3^i - K_3 \cdot z_3 + z_2 \cdot V \cdot \sin(x_2 - x_4) \tag{56}$$

Finally, the resulting error dynamics obtained from equations (54,55,56) is written as,

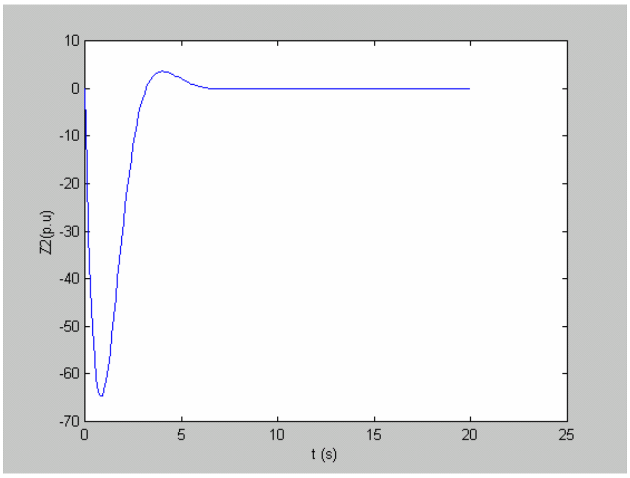
$$\begin{bmatrix} \dot{z}_1 \\ \dot{z}_2 \\ \dot{z}_3 \end{bmatrix} = \begin{bmatrix} -K_1 & 1 & 0 \\ -1 & -K_2 & 1 \\ 0 & -1 & -K_3 \end{bmatrix} \cdot \begin{bmatrix} z_1 \\ z_2 \\ z_3 \end{bmatrix} + \begin{bmatrix} 0 & 1 & 0 \\ -1 & 0 & -V \cdot sin(x_2 - x_4) \\ 0 & V \cdot sin(x_2 - x_4) & 0 \end{bmatrix} \cdot \begin{bmatrix} z_1 \\ z_2 \\ z_3 \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ -\sum_{i=0}^{3} \tilde{a}_i \cdot x_3^i \end{bmatrix} \tag{57}$$

6.5 Identification Procedure

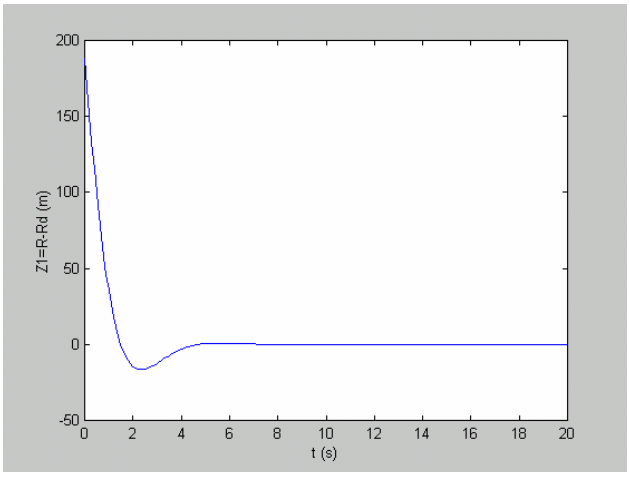
The identification objective is based on the realization of the following tasks:

i) The performing of the turning circle test and the determination of the effective turning radius for a particular rudder angle and the ship speed. In the ship that has been tested, the turning radius was 228 m, with a normal ballast condition and maximum speed 22.68 knots. Other no fundamental features for this identification purpose were a transfer 243 m, advance 334 m and tactical diameter 416 m.

ii) The application of the identification algorithm based on the im-



(a)



(b)

Fig. 9.Temporal variation of the error variables.(a) z_1, (b) z_2

plementation of the equations (54,55,56) and the updating laws (52.a-d) with a suitable simulation program, jointly with the stabilizing functions (37) and (44). The initial values of the estimated parameters $\hat{a}_i\ (i = 0 \cdots 3)$ are assumed to be 25% of the true values.

iii) The application of some optimisation criteria that let us to reduce the z_i $(i = 0 \cdots 3)$ variables to zero. The procedure is based on the Powell optimisation algorithm (Darnell *et. al.*, 1999), a backward Euler integration method with a stepsize of 0.01 s. The gains that were obtained are shown in Table .

The identification algorithm can be extended to the determination of the coefficient c_0. There is an alternative form of computing this coefficient. The procedure starts from the steady value in the yaw rate temporal variation shown in Fig. 5.b. Under this condition, from equation (8.b) yields

$$c = -\frac{\left(a_0 + a_1 \cdot \bar{r} + a_2 \cdot \bar{r}^2 + a_3 \cdot \bar{r}^3\right)}{\delta} \tag{58}$$

where $\bar{r} = 2.93902 \cdot 10^{-2}$ rad/s, $\delta = 0.1745$ rad and a_i (i=0..3), d are the true values shown in Table 3.

Figs.9.a-b,and 10, show that state error variables z_1, z_2 and z_3, used in expressions (34.a,34.b,39), converge quickly towards the null values for the gain values K_1,K_2, K_3 , $\gamma_i\ (i = 0 \cdots 3)$ shown in Table 6.

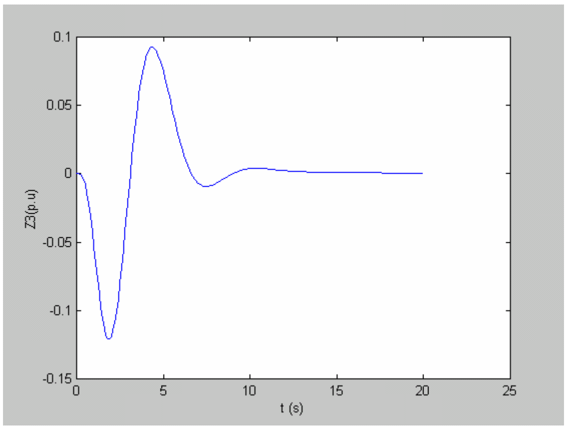


Fig. 10. Temporal variation of the error variable z_3

Table 6. Gains used during the adaptation process

Gains	(p.u)
K_1	1.001
K_2	0.818
K_3	5.639
γ_0	-1
γ_1	-0.9986
γ_2	-10.1722
γ_3	-97223.9
γ_4	$-6.2783.10^6$

Conclusions

The procedure shown in the paper based on the backstepping procedure and under the tuning functions design is capable of realising the adaptive tracking and the identification of this ship model parameters departing from a relative complex ship model dynamics based on the consideration of the nonlinear characteristics of the system.

References

Bech, M. I., Wagner Smith, L., 1969. Analogue Simulation of Ship Maneuvers. *Technical Report Hy-14*., Hydro- and Aerodynamics Laboratory, Lynby, Denmark.

Crane, C.L., Eda, H., and Landsberg, A. 1990. Principles of Naval Architecture. The Society of Naval Architecs and Marine Engineers.

Darnell, P.A.., and Margolis, P.E. 1990. C: A Software Engineering Approach. Springer- Verlag.

Davidson, K.S.M., and Schiff, L.I. 1946. Turning and Course Keeping Qualities. Transactions of SNAME, Vol.54.

Flannery, B.P., W.H. Press and W.T. Vetterling.1989. Numerical Recipes, The Art of Scientific Computing. Cambridge University Press.

Fossen, T.I. 1994. Guidance and control of ocean vehicles, Wiley: Chicester, England.

Gerritsma, J. 1980. Scheepsbewegingen en Sturen II, Report no. 487-K, Ship Hydrodynamics Laboratory, Delft University of Technology.

Journée, J.M.J., and Pinkster, J.A. 2001. Ship Hydromechanics. Delft University of Technology.

Kanellakopoulos. I, Kokotović P., and Morse S. 1991.Systematic design of adaptive controllers for feedback linearizable systems. IEEE Transactions in Automatic Control, Vol.19, 1991, pp. 1241-1253.

Krstić M. , Kanellakopoulos, I., and Kokotović, P. V. 1992. Adaptive nonlinear Control without overparametrization. Systems Control Lett., Vol.19, pp. 177-185.

Krstić, M,. Kanellakopoulos, I., and Kokotović, P. 1995. Nonlinear and Adaptive Control Design, Wiley, New York.

LaSalle,J.P. 1968.Stability theory for ordinary differential equations. J. Diff. Eqs.Vol. 4,1968, pp.57-65.

Navantia. 2001. State shipyard localiced in Cádiz bay. Construction code H81.

Nomoto, K.G, Tagachi, K., and Honda, T. 1957.On the steering quality of ships. International Shipbuilding Progress , Vol. 4.

Norrbin N.H. 1970. Theory and Observations on the Use of a Mathematical Model for Ship Maneuvering in Deep and Confined Waters.Proceedings of the 8th Symposium on Naval Hydrodynamics, Pasadena, CA.

Yoshizawa,T. 1966. Stability theory by Liapunvo's Second Method. The Mathematical Society of Japan.

CHAPTER 5

Models for the simulation of the vehicle movement in ship-port operations.

J. LOZANO ROGADO, A. LÓPEZ PIÑEIRO and R. ATIENZA MARTÍN

[1]Lab. Electrotecnia, Electrónica y Sistemas. Escuela Técnica Superior de Ingenieros Navales. Universidad Politécnica de Madrid.
jesus.lozano@upm.es

Several simulations have been used in naval engineering. Design of ramps is a subject of great importance in ships. In most cases, it is based on previous experiences of designers and not on using an established methodology. This design has a direct impact on the loading and unloading operations and on the port turn-a-round time. In this paper, a macro and a micro model is presented in order to analyze and study the movements of trucks along the ship ramps. This model has been developed from observation and analysis of loading and unloading processes in ships. It also includes several aspects like vehicle and obstacle separations, slope and transitional zones of ramps, movement of nearby vehicles, and anticipation effects of drivers. This model has been programmed in Visual Basic over Microsoft Excel®. A visualization program has also been developed in order to provide convenient results. Using this model, a comparison among several alternative ramp designs can be performed in order to improve the loading and unloading processes in ships.

1 Introduction

In the last years, there is a growing necessity of Short Sea Shipping in the European Union. It is based on an intermodal transport structure of high efficiency, agility and reliability, in which the sea component are the container and RO-RO ships.

Short Sea Shipping is a successful mode of transport in Europe. For instance, in the 1990's it was the only mode that was able to keep pace with the growth of road transport. It has in fact started to outpace road transport.

Short Sea Shipping is also an obvious choice to play a key role in reaching the objectives of the European Transport Policy. It can help curb the forecasted substantial increase in heavy goods vehicle traffic, rebalance the modal shares, bypass land bottlenecks, and it is safe and sustainable.

The European Commission has an active policy to promote Short Sea Shipping. In 1999 it presented a Communication with a comprehensive approach to increase the use of the mode. Furthermore, the recent European Commission White Paper on European Transport Policy for 2010 emphasised the role of Short Sea Shipping in maintaining an efficient transport system in Europe now and in the future.

The Commission is convinced that co-ordinated efforts at all levels (Member States, regional, local, industry and Commission) will substantially help accelerate the growth of Short Sea Shipping, alleviate obstacles and allow Short Sea Shipping to become a true success story of the 21st century.

In all these cases, the load and unload operations are a key element in the process, and in many cases they are the bottleneck of the process. In case of containers, an extensive study has been made both for the process modelling and for the use of high capacity automated facilities, but for the vehicle movements, there is a human component (driver) that makes difficult the analysis methodology.

Our research group has studied in other project the vehicle movements in ships, detected a high interest to study the movements of other types of loads (containers and pallets, etc.) in RO-RO ships and its coordination with the port and hinterland operation. It is necessary for that to develop a formal model of the operations flow suitable for being used in computer applications. In 2005 a new R&D work line has been started in order to develop this concept. The main objectives are:

- To optimize the current flow.
- To improve the design of port facilities.
- To improve the ships distribution, specially the access systems.
- To improve the safety and security of the port's operation.

The vehicle operation in a port, for loading or unloading in ships, can be modelled like series of manoeuvres, which define the vehicle trajectory from the starting to the arrival position. During this process, most of the manoeuvres are standard, so the trajectory can be defined as the sum of "standard manoeuvres" available in a database. Each manoeuvre can be defined as a set of simple geometry sections, basically straight lines and circles.

On the other hand, the model must include aspects of reaction to fixed (scenario geometry) and mobile (other vehicles) obstacles. As should be normal in multilevel models, our general model, MTT (Type Trajectories Model), has been divided into three levels:

- Strategic: in which a route is chosen, avoiding fixed obstacles using a global optimization approach.
- Tactical level: in which the detailed trajectory is defined, checking its feasibility and making local optimizations.
- Operative: in which the vehicle speed in each instant is obtained bearing in mind its observation environment.

The design of external (access to port) and internal (communication between decks) ramps is a subject of great importance in the ships that carry wheeled loads (RO-ROs, RO-PAX and ferries) due to its direct impact on the loading and unloading operations and on the harbour staying time. This aspect has a special attention from the early ship design, participating shipyard engineers, project office personnel and shipowner specialists, with the collaboration of other people like deck equipment suppliers.

The design of the ramps, specially the choice of their number, type, position and geometry (width, length and slope), is based on the previous experience of the designers and in the study of design made in other similar ships but it's a lack of detailed publications about the design methodology.

For that reason, we have developed a model, tested on a computer application that allows analyzing and studying the movements of trailers (truck with semi trailer) along the ship ramps. The type of vehicle chosen corresponds to the most used one in the European transport and it involves more than 80% of the industrial load in RO-RO ships.

The design of the ramps includes both the internal (communication between decks) and the external ramps (ship entry from the quay). The external ramps used to be of "watertight door ramp" type and can be situated in bow, in stern or in one side. In figures 1 to 3, the different types or ramps more used are shown.

Fig. 1. Bow ramp

Fig. 2. Stern ramp

Fig. 3. Side ramp

Internal ramps can be fixed (fig. 4) or mobile (fig. 5) and they usually are located in the ship sides.

Fig. 4. Fixed ramp.

Fig. 5. Internal mobile ramp.

The number and position of external ramps is related to the possibilities of mooring in the quay, the flexibility of operation (the optimal is the bow and stern disposition that allow a vehicle flow always ahead, mooring in each harbour in an alternative way) and the security (the bow ramps are worst because of the risk of water entrance).

By that, in early design stages of the ship, the position of the ramps usually is decided. The present trend is the installation of one or more stern door ramps.

This work is located into the work line of the Naval School (ETSIN) of the Universidad Politécnica de Madrid (UPM) in the analysis and simulation of rolling loads in ships and port installations (ASROBIP), starting with SIFBUP Project (López 2005a and López 2005b), followed by doctorate studies of our R&D group (Atienza 2005).

2 Modelling Approach

A macro-model is enough for a general study of the loading/unloading process, e.g. the phase and zone model used in the SIFBUP application. But the utilization of micro-models is more convenient for a detailed study of the critical phases, in which several detailed aspects like widths, slopes, parking in corners must be represented.

Several applications and models have been developed for the study of vehicle movements in streets and roads (Kesting 2006, Treiber 2000, Chanca and Castellanos 2004) and parking (Maravall 2004, Gomez-Bravo 2002), but these are not of direct application to study manoeuvring at low speed and in complex geometries because they are based on lane movements.

The Intelligent Driver Model (IDM) developed by Treiber (Treiber and Treiber 2000) has been used as a start point for the development of a new micro model. The model is based on the following acceleration equation:

$$\frac{dv}{dt} = a\left[1 - \left(\frac{v}{v_0}\right)^{\delta} - \left(\frac{s^*}{s}\right)^2\right] \tag{1}$$

Where:

- a is the “comfortable” maximum acceleration.
- v is the current speed.
- v_0 is the target speed when driving on a free road.
- s^* is the minimum distance between vehicles.
- s is the distance to the front vehicle
- δ is the acceleration exponent (constant)

The first two terms in the square brackets represent the behaviour in the present of speed changes and the other term express the braking effect when there is another vehicle in front of it. This model doesn't bear in mind the available width, the floor slope or the state of road surface, which have a lot of importance in our case.

From the observation and analysis of loading and unloading processes in ships, several aspects not included in IDM can be emphasized:

- The desired velocity is smaller than the velocity on roads and lower in ship decks than in quays.
- The minimum separation with the front vehicle or obstacle is different if our vehicle is moving or parking.
- The side distance has an important influence in the velocity if it falls from a minimum threshold. The velocity also falls in the case of a low height of the roof. A minimum and maximum width can be considered; between these values a lineal function of velocity can be supposed.
- In ramps it is necessary to distinguish between the constant slope and transitional zones. The higher is the slope the lower is the velocity. The loss of velocity is increased in the entrance and exit zones by the discontinuity effect in the slope.
- The effect of movement of other vehicles in the proximity affects in a different way the front and the lateral vehicles (both side and forward distance must be studied). In the case of velocity reduction caused by ahead vehicle (see figure 6), the more significant parameter is the gap between them "S". In the case of side vehicles, the reference parameter is the length distance "ΔX".

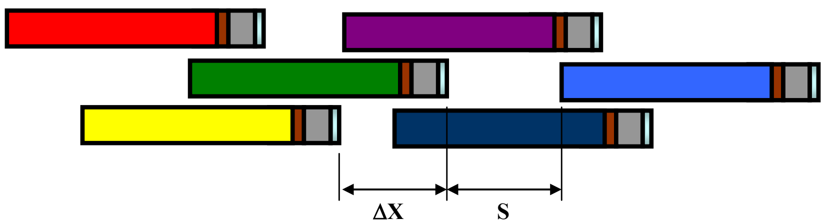


Fig. 6. Close vehicles effect

Once the vehicle trajectory and scenario geometry are known, applying the previous concepts, to obtain the velocity profile in function of position results immediately. But drivers realize anticipation effects in the presence of velocity reduction in order not to enter in critical zones braking suddenly and they keep the limited velocity until most of the vehicle has passed it. In figure 7 is shown an example where H is the ramp profile, Vpo and Vp are the velocities without and with the anticipation effect.

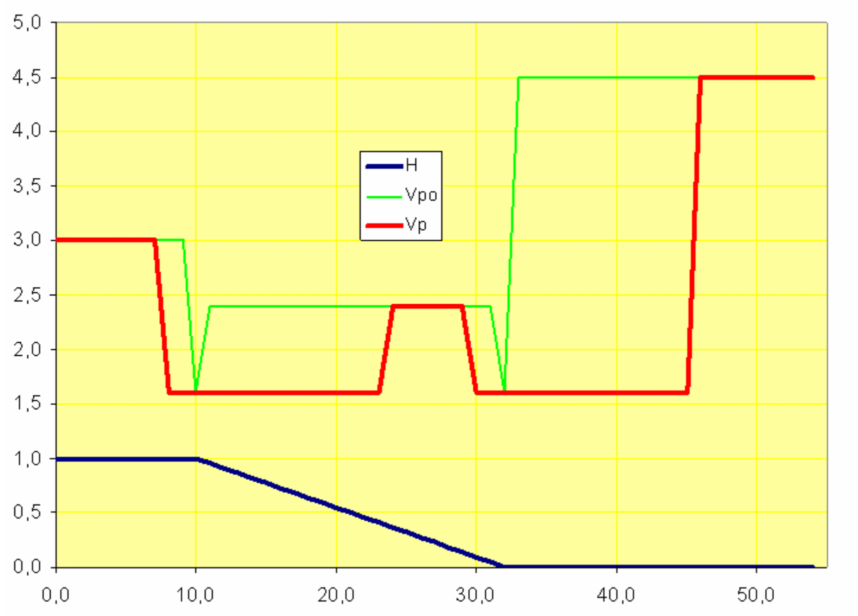


Fig. 7. Truck velocity profile on a ramp

Bearing in mind these aspects, a length movement's micro-model in ships has been developed (M3LEB model). It is based on vehicle trajectories that can be predicted previously. Into the ships, (see figure 8) several lines are marked to delimit the parking positions, which can be used to define the trajectories.

The following situations can be observed in the vehicle trajectories:

- To coincide for the vehicles parked in the same lane.
- To interfere with other if the distance is smaller than the vehicle width.
- To approximate if correspond to the left or right of the own lane.
- Being independent in the remaining cases.

In case one and two, vehicles must drive with a minimal length distance and in the third case, if there is overlapping between vehicles, its velocity falls.

Fig. 8. Lines of parking positions used to define the trajectories.

2.1 General Equation of the Model

After several trials and simulation experiments, it has been established a mathematical formulation for the model, similar to IDM, based on the following equation:

$$A = A_0 \cdot \left[1 - \left(\frac{V}{V_0} \right)^4 \right] \tag{2}$$

Where:

- A is the current vehicle acceleration.
- A_0 is the maximum operational acceleration.
- V is the current velocity.
- V_0 is the desired velocity.

But in M3LEB model, V_0 isn't constant. It is the minimum of the velocities limited by the scenario geometry, which have been divided into the following:

- V_p: velocity limited by the geometry; the values (time independent) are calculated for every point of the vehicle trajectory. The type of zone (deck or quay), the width clear with the ship structure and the slope and points of discontinuity, are taken into account in this calculus.
- V_d: velocity limited by the gap with front vehicle, using a formula derived from the IDM. It is a function of the length distance of the vehicles and time variable.
- V_W: velocity limited by the near side vehicles, which drive in the nearby lanes. It is necessary four possible vehicles to be considered (front, back, by the left and by the right). Its value depends on the length gaps between them and the lane width (or trajectory distances).

It has been developed a simulation application in order to study the unloading process of the first trucks. This application is limited for trucks moving forwards, simulating in a detailed way the behaviour in the presence of ramps, bearing in mind both its slope and width.

3 Simulation

An application for simulating the unloading of the first vehicles (SPDV) has been performed. The main objectives of the SPDV application are the following two:

– To have a prototype for the first adjusts and validation of the M3LEB model, allowing testing the concept and equations structure of the model and the interaction method of each vehicle with its environment into the operative model.
– To have results of the movement of vehicles in ramps, a critical point in the method developed to the ship's ramp design.

To fill these objectives, it is necessary to have a detailed simulation of both the first vehicles unloading and the loading of the last ones. The SPDV application is able to simulate the unloading, but the loading process has not carried out because, by the moment, we haven't numeric data enough of reverse movements of trucks.

3.1 General Structure of the Program

The application is developed using Visual Basic over MS Excel with the aim of simplifying the data entry and results output, automating most of the calculations required.

It is designed to simulate the forward movement of trucks, although it could be adapted to other vehicles and directions, changing the equations parameters, starting from a movement following predefined lanes although lane changes are allowed. The maximum number of vehicles to be processed is 100, although using more than 30 is not recommended.

In a principal sheet of the Excel book, the name of the ship and the study must be filled. Before making the calculations, the ship geometry and the velocity profile desired tied to the position (Vp) in an auxiliary sheet called “Datos”. As results of the simulation, the position (into the ship and on the quay) of each vehicle is obtained with a 1 second resolution and shown in a table and in an evolution graphic and also a summary with the most significant results.

3.2 Animated Presentation

The results of the simulation can also be presented in an animated way, by using a postprocessing application in which can be generated some videos in 2D and 3D to analyse intuitively and provide a whole view of the unloading process. In figure 9 and 10 samples of the 2D and 3D visualization are shown.

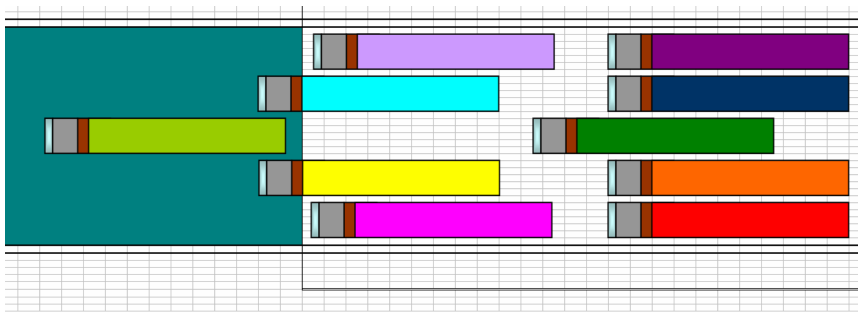


Fig. 9. 2D visualization

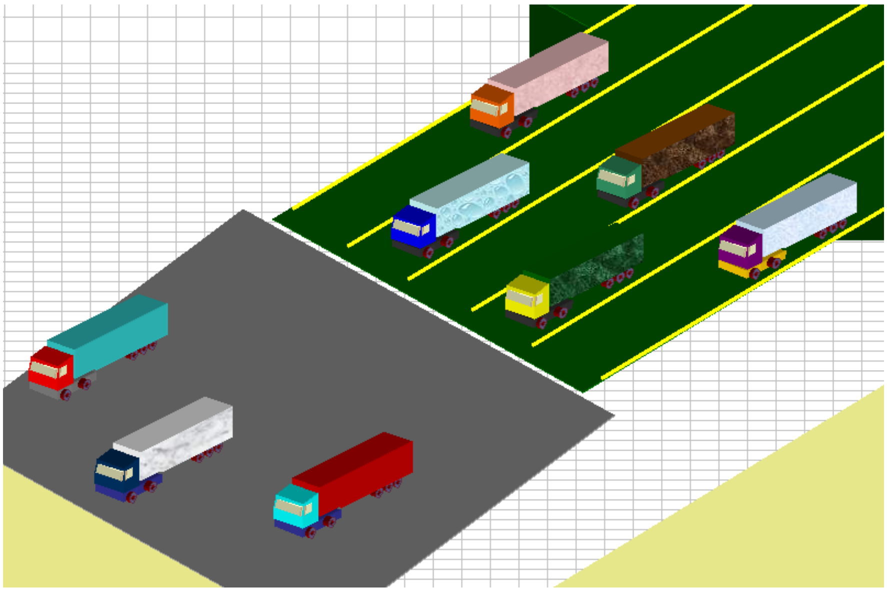


Fig. 10. 3D visualization

Also we use this facility to debug the model and program details, to validate it quantitatively and to allow the intuitive presentation of disposition and operation alternatives. This application has also programmed in Visual Basic over MS Excel.

4 Results and Discussion

To show the use of these tools we study a ship with two stern ramp configurations and analyze the unloading process. In figure 11 is presented the first alternative with one stern ramp and a mobile ramp between decks 2 and 3.

The ship can hold up to one hundred and twenty vehicles in the following distribution:

- Deck 1 (hold): 6 + 7 + 5 = 18 platforms
- Deck 2 (main): 17 + 16 + 17 = 50 trucks
- Deck 3 (upper): 19 + 18 + 3 + 10 + 2 = 52 trucks

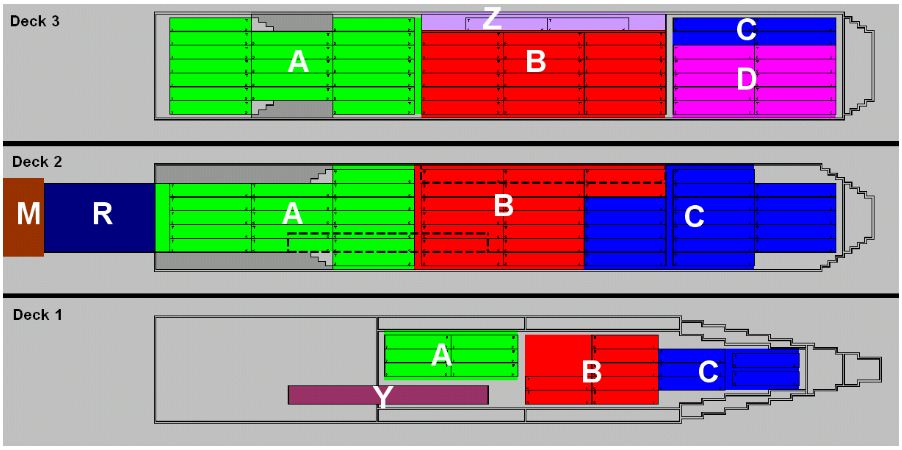


Fig. 11. Cargo plan. First distribution

The loading process begins opening the stern door-ramp, the unloading of vehicles of deck 2, next lowering the deck-3 ramp, unloading vehicles from deck 2 and 3, then opening of deck 1 ramp and unloading the platforms using port tractors (mafis). The loading process is similar to unloading but in an inverse way.

The results of the first vehicle unloading simulation using SPDV application is shown in figure 12. It can be observed the interaction effect between vehicles, especially at the moment to arrive at the exit ramp.

Comparing the results of micro and macro models applications (SPDV and Sifbup) can be obtained different average time of journey: 46,5 and 62,1 second respectively, but it is due to the time of exit of the parking place in Sifbup (15 seconds). If this extra time is discounted, the average time obtained is 47,1 second. In this case, a good correlation between the models of two applications is obtained, taking Sifbup a higher security margin.

Something similar happens with vehicles flow because in macromodel, the lashing time has taken into account, and the overlapping between vehicles at the moment of passing a zone is less optimized.

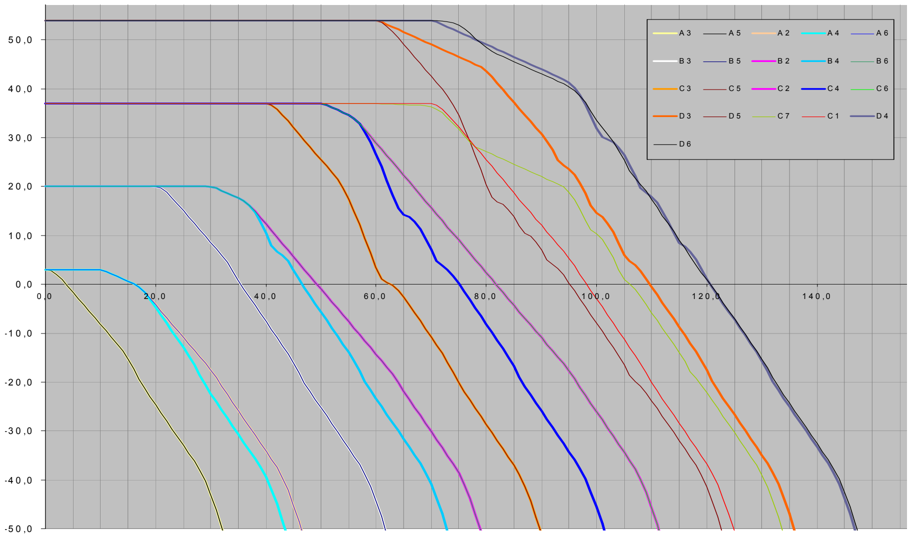


Fig. 12. Unloading first 21 vehicles simulation.

In the second distribution presented in figure 13, there are two stern door ramps. The left (R) is connected with a fixed ramp to the upper deck.

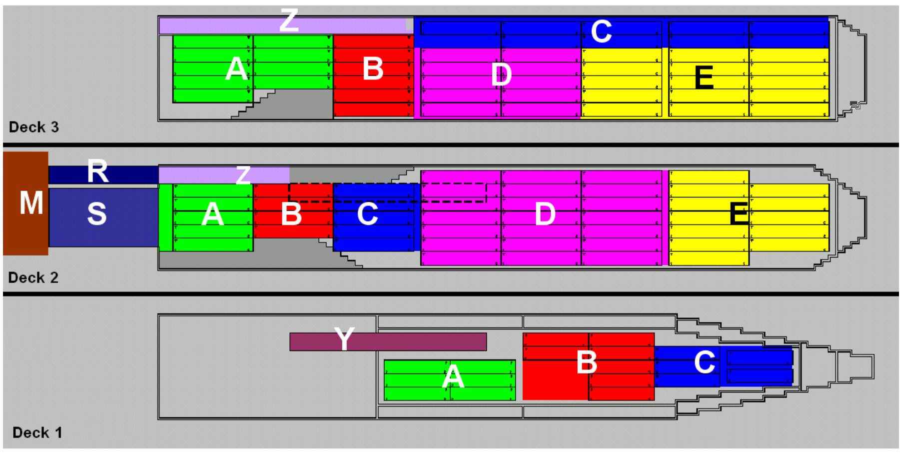


Fig. 13. Cargo plan of second distribution

The loading and unloading process is similar to the first distribution but in this case, the loading and unloading of decks 2 and 3 are independent.

In table 1, a summary of results obtained with the two alternatives is shown. It can be observed that the first alternative has less charge capacity (6 vehicles) but the time is 20 minutes short, so, the first option is preferable for short voyages and the second one for long ones.

Table 1. Comparison of results obtained in simulation

Option	**N. of vehicles**	Unloading time	Platform load./unl. time	Total loading time	**Load./unl. time**	20 vehicles time	Av. Unloading 20 veh. time	Av. wait. time at loading
1	**102+18**	38,9'	60'	51,2'	**2 h 30'**	147 s	86 s	25,5 s
2	**96+18**	30,3'	60'	39,9'	**2 h 10'**	120 s	73 s	15,8 s

Finally, in figure 14, a comparison of the exit time of the first 20 vehicles is shown. The effect of independent ramp in second option can be observed. This second option also can be improved allowing vehicles of upper deck to go down when port door is opening.

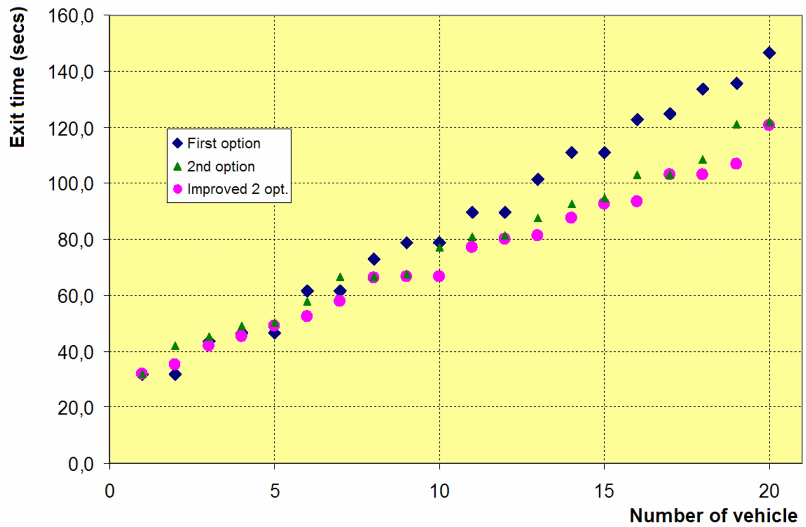


Fig. 14. First 20 vehicles exit comparison.

Conclusions

The first conclusion is that the utilization of formal elements for the analysis of vehicles on board loading and unloading is possible and moreover the realization of studies and simulation for the analysis of alternatives for the ramps arrangements and operation procedures.

A methodology for the decision making on the number, size, type and location of ramps that will permit optimizing the arrangement of loaded charge ships in function of their traffic has been successfully developed. The macro and micromodel applications have been tested for the movements of vehicles on board.

This paper shows a step in a workline. Future works are the following:

- Data taking for the calibration and validation of forward movement.
- Study of the special features of reverse movements. Data taking.
- Definition of complex maneuvers. Integration of models.
- Study of vehicle movements on ports.
- Development of a more friendly tool, user oriented.

References

Atienza R. 2005. Trabajo tutelado “Generador de rutas óptimas”. A. “Modelos de simulación de personas y vehículos”. E.T.S.I.N. Madrid

Alvariño R. “El proyecto preliminar del ferry Ro-Ro. Ingeniería Naval. Madrid, mayo 2005; octubre 2005.

Chanca, J.A. and Castellanos, J.A. 2004 “Simulación Microscópica De Tráfico Urbano Y Su Aplicación En Un Área De La Ciudad De Zaragoza” Actas de las XXV Jornadas de Automática. Universidad de Castilla La Mancha.

E.T.S.I.N., Trasmediterránea and Izar. SIFBUP-V v1.1. Manual de Usuario. Madrid, enero 2005.

Gómez-Bravo F., Cuesta F. and Ollero A. 2002 “Aparcamiento paralelo en vehículos no holónomos con y sin remolque” Actas de las XXIII Jornadas de automática. Universidad de la Laguna.

Kesting, A.; Treiber, M. and Helbing, D. 2006 “Game-theoretic approach to lane-changing in microscopic traffic models” Elsevier science.

López A. et al. 2005. “Simulación del flujo de personas y vehículos en buques de pasaje. El proyecto SIFBUP”. Ingeniería Naval. Madrid.

López A. “Proyecto ASROBIP. Modelo trayectorias tipo” E.T.S.I.N. Madrid, julio 2005.

Maravall, D., Lope, J. de, Patricio, M.A. 2004 “Coordinación Competitiva de Objetivos en el Aparcamiento Automático”, Actas del Tercer Congreso Español de Metaheurísticas, Algoritmos Evolutivos y Bioinspirados, Córdoba 4-6 febrero 2004, pp 43-50.

Treiber M. “Longitudinal trafic model: The IDM” available at: http://141.30.186.11/~treiber/index.html

Treiber, M.; Hennecke, A. and Helbing D. 2000 "Congested Traffic States in Empirical Observations and Microscopic Simulations" Phys. Rev. E 62, 1805–1824.

CHAPTER 6

Reactive control of a visually guided underwater cable tracker

J. ANTICH, A. ORTIZ, and G. OLIVER

Department of Mathematics and Computer Science
University of the Balearic Islands
Cra. Valldemossa, km 7.5 - 07122 Palma de Mallorca (Spain)
{javi.antich,alberto.ortiz,goliver}@uib.es

This chapter presents a behavioural control architecture for locating and tracking an underwater cable/pipeline on the basis of a computer vision algorithm. The whole system is validated by means of a simulator named $NEMO_{CAT}$. Missions of different complexity are considered, favouring, most of them, that the local minima problem arises by adding obstacles forming troublesome configurations such as U-shaped canyons. Despite this fact, the vehicle successfully detects and follows the cable/pipeline while avoiding the above-mentioned obstacles for all the intended scenarios. Nevertheless, the proposal presents some limitations due to its reactive character. Apart from a brief discussion at the end of this chapter, they can be found identified and solved from a generic point of view in (Antich, 2006).

1 Introduction

1.1 Task Relevance

The feasibility of an underwater installation, such as power or telecommunication cables and pipelines, can only be guaranteed by means of a suitable inspection programme. This programme must provide the company with information about potential hazardous situations or damages caused by the mobility of the seabed, corrosion, or human activities such as marine traffic or fishing. Nowadays, the surveillance and inspection of these installations are carried out using video cameras attached to ROVs normally controlled by operators from a support ship. Obviously, this is a tedious task because the operator has to concentrate for a long time in front of a console, which makes the task highly prone to errors mainly due to loss of attention and fatigue. Besides, undersea images possess some peculiar characteristics which increase the complexity of the operation: blurring, low contrast, non-uniform illumination and lack of stability due to the motion of the vehicle, just to cite some of them. Therefore, the automation of any part of this process can constitute an important improvement in the maintenance of such installations with regard to errors, time and monetary costs.

1.2 Contents of the Proposal

The special visual features that artificial objects have allow distinguishing them from the rest of objects present in a natural scenario even in very noisy images. In our case, the rigidity and shape of an underwater cable/pipeline can be exploited by a computer vision algorithm to discriminate it from the surrounding environment. This fact makes feasible the automatic guidance of an AUV by means of visual feedback to carry out maintenance/inspection tasks. Following this strategy, a novel approach to the problem of detecting and tracking an underwater cable by analysing the image sequence from a video camera attached to an ROV was described in (Ortiz *et al.*, 2002).

In this chapter, a new version with similar success rate, better performance and lower complexity is discussed. Furthermore, the control architecture for locating and tracking the cable autonomously on the basis of the above-mentioned vision system is also presented.

The vision subsystem has been tested using sequences from a video tape obtained in several tracking sessions of various real cables with an ROV driven from the surface. These cables were installed several years ago, so that the images do not present highly contrasted cables over a sandy seabed; on the contrary, these cables are partially covered in algae or sand, and are surrounded by algae and rocks, making thus the sequences highly realistic. The mean success rate that has been achieved is about 90% for a frame rate of much more than 25 frames/second.

On the other hand, the control architecture is reactive in order to produce timely robotic responses in a dynamic, unstructured and, in general, unknown world such as the submarine. The motor schema methodology has been adopted in its implementation (Arkin, 1989). Additionally, the behaviour responses were encoded using potential fields (Khatib, 1986). Simplicity, modularity and good performance in quite complex navigation tasks are the most important characteristics of the resultant behavioural architecture. It will be shown, however, that the inherent shortcomings of reactive approaches were also inherited, limiting thus the application field of the proposal. A continuation of this work providing a generic solution to such problems can be found in (Antich, 2006).

The whole system has been validated using the simulator $NEMO_{CAT}$ described in (Antich and Ortiz, 2004).

2 The Vision System

2.1 Outline

Artificial objects usually present distinguishing features in natural environments. In the case of the cable, given its rigidity and shape, strong alignments can be expected near its sides. The vision system which is proposed precisely exploits this fact to find the cable in the images.

In order to obtain the cable parameters, the system splits the image to be analysed in a grid of cells which are processed separately. This division intends to reinforce the evidence of the cable in those areas of the image where it appears clearly defined. Different steps are carried out to locate the cable in every cell of the resultant grid. First, an optimised segmentation process is executed to find image regions as approximated as possible to the scene objects. Given the contours of such regions, alignments of contour pixels are determined. If among those alignments there is strong evidence of the location of the cable (mainly two alignments with a great number of pixels lined up and with a high degree of parallelism, even without discounting the perspective effect), then the cable is considered to have been located and its parameters are computed. After analysing all the cells of the grid, the partial results obtained are merged to achieve a global agreement about the real cable position and orientation in the image. By way of example, figure 1 shows the cable detection process for a real image.

Once the cable has been detected, its location and orientation in the next image are predicted by means of a Kalman filter, which allows reducing the pixels to be processed to a small ROI (Region of Interest). In this way, the computation time is considerably lowered together with the probability of misinterpretations of similar features appearing in the image.

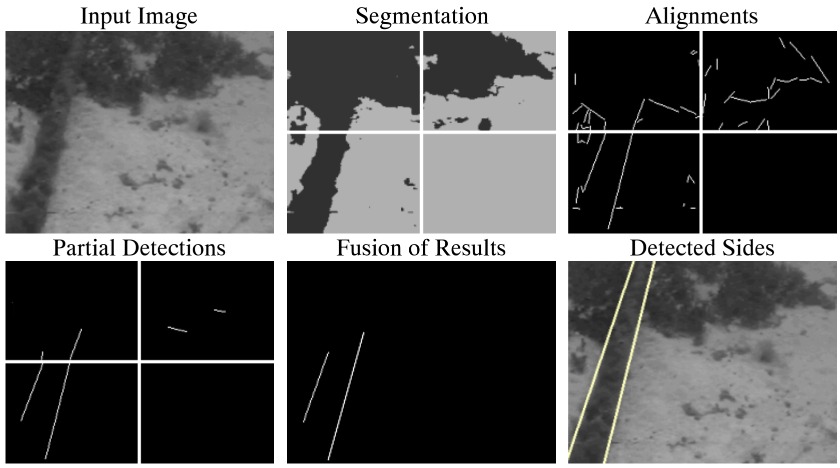


Fig. 1. Intermediate and final results for a real image split in 2×2 cells

When tracking the cable, a low or null evidence of its presence in the ROI can be obtained. In such a case, the image is discarded and a transient failure counter increased. If this anomalous situation continues throughout too many images, then it is attributed to a failure in the prediction of the ROI, resulting in two special actions: the Kalman filter is reset and the ROI is widened to the whole image. In other words, in front of a persistent error, the system will no longer make use of the knowledge acquired from the last processed images.

2.2 The Segmentation Process

A gray-level thresholding technique has been applied to carry out the segmentation of every grid cell. It is based on a particular histogram where the relevant objects of the scene can be more easily distinguished than using the traditional gray-level histogram.

In order to obtain the mentioned histogram, the cell of the grid to be analysed is first transformed into the {gray-level, gradient modulus} space. This transformation consists in building a bidimensional histogram where one horizontal axis corresponds to gray-level, the other horizontal axis corresponds to a digital approximation of the modulus of gray-level gradient —the Sobel operator has been used—, and for every combination {gray-level, gradient modulus} the vertical axis is the number of pixels in the cell having that gray-level and that gradient modulus.

In the case of several objects with different gray-levels, the ideal bidimensional histogram should look like figure 2. In effect, if the image can be approximated by a noisy piecewise constant bidimensional function, the interior of any object in the cell has gradient near zero, so that pixels in the interior zones are located in the lower part of the histogram, with regard to gradient. Border pixels among objects, however, are located in zones of higher gradient, joining the clusters corresponding to the interiors of such objects in a "fingers"-like fashion.

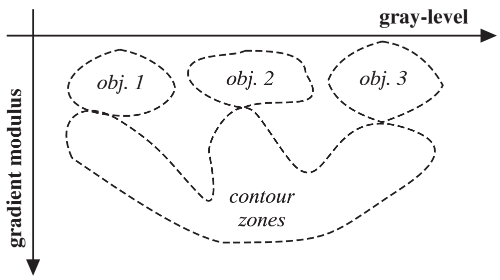


Fig. 2. Ideal bidimensional histogram

Once the bidimensional histogram has been built, it is projected onto the plane {gray-level, number of pixels}. The projection is cumulative and does not consider the pixels whose gradient is greater than a predefined threshold. Ideally, this parameter should reject those pixels that belong to the contour zones because they blur the separation among the objects that appear in the cell with different gray-level. In this way, it is expected that the resultant one-dimensional histogram can be successfully managed by a simple and low-time consuming segmentation technique such as thresholding. The value of the mentioned parameter is indirectly specified by means of a percentage that indicates the amount of pixels that have to be projected.

The next step partitions the grid cell into the regions that can be intuitively distinguished in the previously computed histogram, looking for its valleys. A new threshold is used again. Gray-levels with a number of pixels lower than that threshold are considered candidates to turn into a valley. This second segmentation parameter is expressed as a percentage of the average height of the histogram. In general, it will be usual to find lots of candidate gray-levels grouped together. In these cases, all of them are treated like a single candidate with a width equal to the number of involved gray-levels.

Once the preliminary list of candidate valleys has been obtained, a first selection is carried out. Candidates whose width is not greater than a predefined value are automatically removed. In this way, untrue valley detections that can occur due to the typical lack of continuity in the gray-level distribution of the objects that appear in real images are filtered. Afterwards, a representative for every candidate that has satisfied the previous constraint is chosen: the midpoint of the gray-level interval covered by the candidate. In this context, the first and the last candidates over the histogram are especially treated, setting their representatives to the gray-levels 0 and 255 respectively.

In this point, the limiting gray-level among the objects, or classes from now on, are clearly defined by means of the representatives of the almost-definitive valley list. A new constraint is applied in order to obtain the final list: the classes that the valleys delimit have to be qualified as relevant. A class is considered relevant when the number of pixels that it comprises is greater than a certain percentage of the total number of pixels of the histogram. Non-relevant classes are automatically removed allocating the corresponding gray-levels to their neighbouring classes.

The definitive valley list and therefore the definitive set of classes are now available. In the best case, there will be as many classes as objects appear in the scene. However, it will only be possible when the gray-levels of such objects differ to one another. Objects with the same gray-level will only be able to be distinguished if they are not adjacent in the image grid cell.

Finally, the segmentation is generated according to the class information obtained so far. It is carried out in a simple way: each pixel of the analysed cell is allocated to the class including its gray-level. Note that the total division of the gray-level range among the detected classes guarantees the complete cell segmentation.

2.3 Detection of the Cable

Once the cell has been segmented, the system proceeds to locate the cable executing the tasks enumerated in figure 3. This step is carried out from the contours of the segmented cell, by looking for lines which can belong or be near the sides of the cable. In this context, a line is defined as a set of connected contour pixels not including branches. On the other hand, it is important to note that, unlike previous versions of the system (Ortiz *et al.*, 2002), the detection step does not assume a vertical orientation of the cable in the image. This restriction is removed in order to use any evidence of the presence of the cable. However, it also increases the probability of erroneous detections.

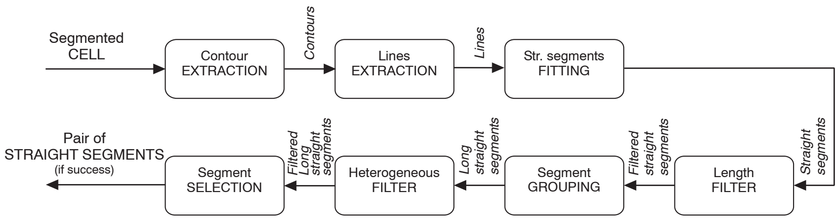


Fig. 3. Flow diagram of the cable detection step

Lines are obtained by scanning the segmented cell from bottom to top. The direction of scanning is important as the lower part of the image tends to be clearer than the upper part when the camera is not oriented towards the seabed, due to the properties of light propagation undersea. Once a contour pixel (i,j) has been found, adjacent pixels are selected according to the preference matrices shown in equations 1 and 2. Numbers indicate preference, the lower the higher, except for 0 which represents forbidden selection. Depending on the expected orientation of the cable in the image, the system uses one matrix or the other. For instance, the preference matrix shown in equation 1 is used when the Kalman filter predicts that the cable is going to appear vertical or near-vertical in the image. In this way, lines in that direction are favoured against horizontal and circular lines. On the other hand, horizontal or near-horizontal predictions of the cable orientation make use of two alternative preference matrices (see equation 2). These matrices give priority to the left (a) and right (b) horizontal following of the contour. Details about the selection criteria of each one of the preference matrices can be found in figure 4. When, for a given contour pixel, there is no adjacent pixel in the preferred directions, the process of tracking the line finishes and a new one starts by resuming the scanning of the cell from the point it was left.

$$\begin{bmatrix} 2 & 1 & 3 \\ 4 & (i,j) & 5 \\ 0 & 0 & 0 \end{bmatrix} \tag{1}$$

$$(a) \begin{bmatrix} 2 & 4 & 0 \\ 1 & (i,j) & 0 \\ 3 & 5 & 0 \end{bmatrix} \qquad (b) \begin{bmatrix} 0 & 4 & 2 \\ 0 & (i,j) & 1 \\ 0 & 5 & 3 \end{bmatrix} \tag{2}$$

A straight segment fitting task follows next. This process can be seen as a low-pass filter to remove noise either due to the redefinition of the cable contours caused by the proliferation of flora on top of and by the

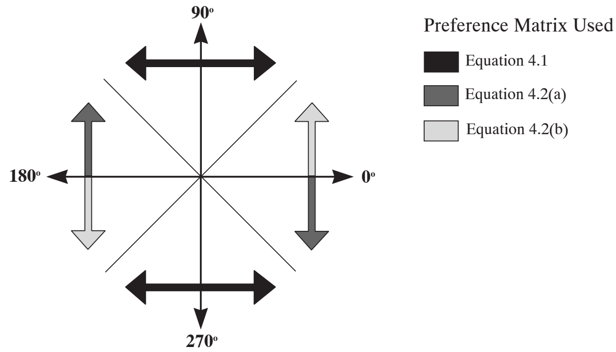


Fig. 4. Preference matrix utilization based on the expected orientation of the cable in the image

cable, and due to the processes of acquisition and segmentation. Total least squares is used in the fitting (Duda and Hart, 1973). As the fitting error can become large in some cases, a control procedure is executed after each fitting. It is as follows:

- For each point p_i belonging to the line L, its orthogonal distance to the fitted straight segment S, $d(p_i, S) \geq 0$, is computed.
- If $d(p_j, S) = max\{d(p_i, S) \mid p_i \in L\} \geq k_e$, where k_e is a threshold, then L is split into two halves by the point of greatest local maximum error which is not an end of the line. In this way, the pathological case of always splitting the line by its ends is avoided. Figure 5(a) shows a typical example.

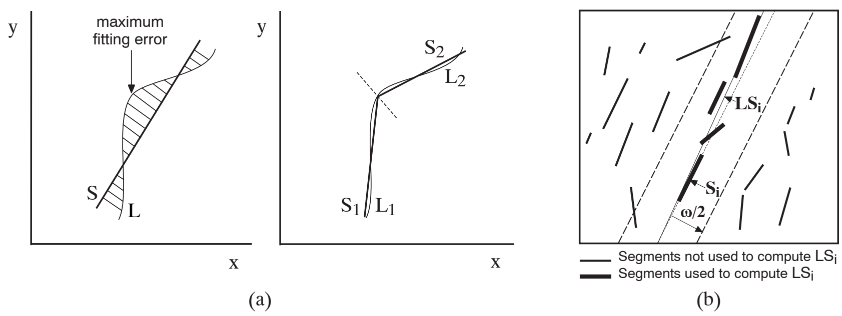


Fig. 5. (a) splitting of a line L; (b) co-linearity analysis

Next, the resultant set of straight segments is filtered according to their length. In this context, the length of a straight segment is defined by means of the total number of contour points that it fits. The filter consists in keeping the N longest straight segments. In this way, it is intended to reduce the size of the problem in a controlled way. Besides, as the segments that supply more information are kept, a non-negative influence of the filter on the results obtained is expected. However, it will depend on the choice of N.

Subsequently, a co-linearity analysis is applied to the set of straight segments obtained, in order to join the segments that can be considered as originally belonging to the same long straight contour. As an example of the analysis performed, consider the set of segments that have passed the length-based filtering process (see figure 5(b)). For each straight segment S_i under analysis, a new long straight segment LS_i is calculated using again total least squares. This time, the points used in the fitting are those contour points corresponding to the straight segments which completely fall within a strip-shaped region aligned with S_i, whose width is the tolerated co-linearity error w.

Immediately afterwards, the resultant set of straight segments is filtered again. Unlike the specific length filter, now each long straight segment is evaluated according to a suitable combination of four different criteria. Those segments that obtain a score lower than a predefined threshold are removed. In this way, it is intended to reject straight segments with little probability of belonging to one side of the cable. The criteria used to assess such segments are length (C_1), fitting error (C_2), average of the gradient modulus of the contour pixels fitted by the straight segment considered (C_3), and, finally, the standard deviation of the differences among the gradient directions of the aforementioned contour pixels (C_4). The partial and normalised assessments obtained of each one of the previous criteria are weighted in order to compute the final one, a_{ss} (see equation 3). Successful results have been achieved assigning a higher weight to the criteria C_1 and C_2. The final assessment will be re-used by the next task.

$$a_{ss} = w_1 C_1 + w_2 C_2 + w_3 C_3 + w_4 C_4 \tag{3}$$

The last task of the detection step consists in choosing the pair of long straight segments which are likely to correspond to the sides of the cable. Considering its morphological characteristics, the task mainly looks for two long and parallel straight lines. Initially, each possible pair of straight segments is checked according to the distance that separates them. Those pairs whose separation reasonably differs from the expected width of the cable in the images are discarded. Note that, using this new parameter, the system assumes that the width of the cable does not change significantly. If a non-zoomed camera system is used, it will be true as long as the AUV is able to keep the distance to the seabed approximately constant. The proposed control architecture incorporates a behaviour that covers that specific task. In this way, the probability of erroneous detections is considerably reduced. Afterwards, three different criteria are used to evaluate each surviving pair of straight segments: degree of parallelism (C_5), average of the Euclidean length of both segments (C_6), and, finally, the average of the individual assessments obtained by such segments in the previous task (C_7). Once all the final assessments, a_{pss}, have been computed (see equation 4), the pair with the highest one is selected. In case the maximum score is below a minimum value, the cell is not considered to contain enough evidence of the cable.

$$a_{pss} = w_5 C_5 + w_6 C_6 + w_7 C_7 \tag{4}$$

2.4 Fusion of Partial Results

Once all the cells of the grid have been processed, each cell contributes to the computation of the global position and orientation of the cable using the resultant partial detections. Those cells for which two long parallel straight segments showing enough evidence of the presence of the cable have been found contribute with that pair. In the remaining cases, the contribution consists in the segments surviving the filtering tasks previous to the pairing. In this way, both sides of the cable are not required to lie in the same cell of the grid so as to be taken into account. Results are merged considering non-overlapping groups of 2×2 cells in a pyramidal way, reducing, at each iteration, the number of cells from $N \times M$ to $\lceil \frac{N}{2} \rceil \times \lceil \frac{M}{2} \rceil$. For every set of cells, the fusion of results is achieved by re-executing the segment grouping, heterogeneous filtering and segment selection tasks previously described (see figure 3 again). The merging process finishes when only one cell is left. In this case, the average of the pair of segments resulting from the segment selection task, if any, constitutes the output of the vision system.

2.5 Cable Tracking Strategy

The tracking stage is based on the hypothesis that the cable parameters are not going to change too much from one image to the next. As a result, once the cable has been detected in the image sequence, the computed position and orientation are used to predict the new parameters in the following image. This prediction allows introducing a further checking point in the sense of comparing predicted with computed parameters. At the same time, the image area of where to look for the cable can be reduced and, thus, the probability of success increased. This area is the above-mentioned Region Of Interest (ROI). As it was said before, in case the system is not able to find enough evidence of the cable in the ROI, the recovery mechanism described at the beginning of the section will be activated.

To predict the cable parameters, the vision system makes use of a linear Kalman filter for the main axis of the cable. Previous versions of the system carried out such prediction by means of two filters, one for every side of the cable (see (Ortiz *et al.*, 2002) for details). The main axis has however shown to be more predictable than the sides. The reason why the linear version of the Kalman filter is used is because in general the dynamics and the motion of the vehicle carrying the camera are unknown, so that, as the simplest option, the linear variant was tested, giving fairly acceptable results. The state vector X contains the position and orientation of the main cable axis in the Hough plane (ρ, θ) as shown in figure 6. The model of the filter is expressed in equation 5, where v and w represent, respectively, the process and the measurement noises.

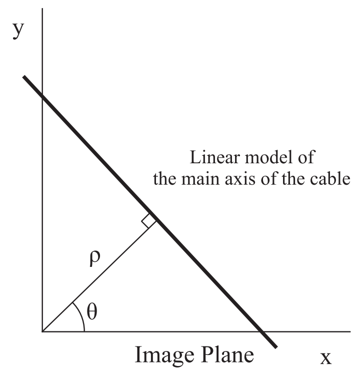


Fig. 6. Predicted main cable axis parameters

$$\begin{aligned} X &= (\rho, \theta) \\ X(t+1) &= X(t) + v(t) \\ Z(t+1) &= X(t) + w(t) \end{aligned} \tag{5}$$

To compute the process noise v, several real sequences were manually analysed and the differences between consecutive frames in the cable positions and orientations were computed. At the end of this procedure, an estimation of the covariance matrix of such noise was available. As for the measurement noise w, the system was faced against noisy synthetic sequences and the deviations between the real orientations and positions of the cable and the measured ones were determined. The corresponding covariance matrix was obtained from those deviations.

Finally, the ROI for the next image is computed as follows: first, the position and orientation of each side of the cable are estimated on the basis of the predicted main axis and the expected cable width in the images; afterwards, a small tolerance factor is added to both sides.

2.6 Performance Assessment

In order to test the vision system, real image sequences coming from several ROV sessions recorded on video tape have been used. Specifically, five sequences were selected from that recorded material to carry out the experiments. Although they are not very lengthy, they cover a wide range of complexity: steep gradient in illumination, low contrast and blurring, objects overlapping the cable, instability in the vehicle motion, etc. Table 1 shows relevant information about every sequence. The success rate appearing in the table refers to those frames for which the ROI wholly includes the cable and the system has been able to determine correctly its location. All the tests were run on an Intel Pentium III 800 MHz machine executing Windows XP Professional, and the resolution of the images was half-NTSC (320×240 pixels).

Figure 7 shows the output of the vision system for an excerpt of the first three real sequences. As can be observed, the system, in general, tends to return the main axis of the cable within the cable region of each image, so that it can be said the position of the cable is correctly detected every time. The orientation measured, however, is affected by the noise present in the image. Consequently, sometimes deviates from the real orientation.

3 The Control Architecture

3.1 Mission Stages and Sensory Equipment

In a typical mission, several different stages can be distinguished: diving, sweeping, tracking and homing. In the first one, the AUV, after having

Table 1. Image sequence results

Sequence	Length (frames)	Frame rate achieved	Wrong detections	Success rate
1	253	61.76 f/s - 16 ms/f	37	85%
2	499	33.02 f/s - 30 ms/f	64	87%
3	386	39.93 f/s - 25 ms/f	11	97%
4	116	33.29 f/s - 30 ms/f	11	90%
5	113	36.75 f/s - 27 ms/f	11	90%
Average	1,367	40.95 f/s - 24 ms/f	134	89.8%

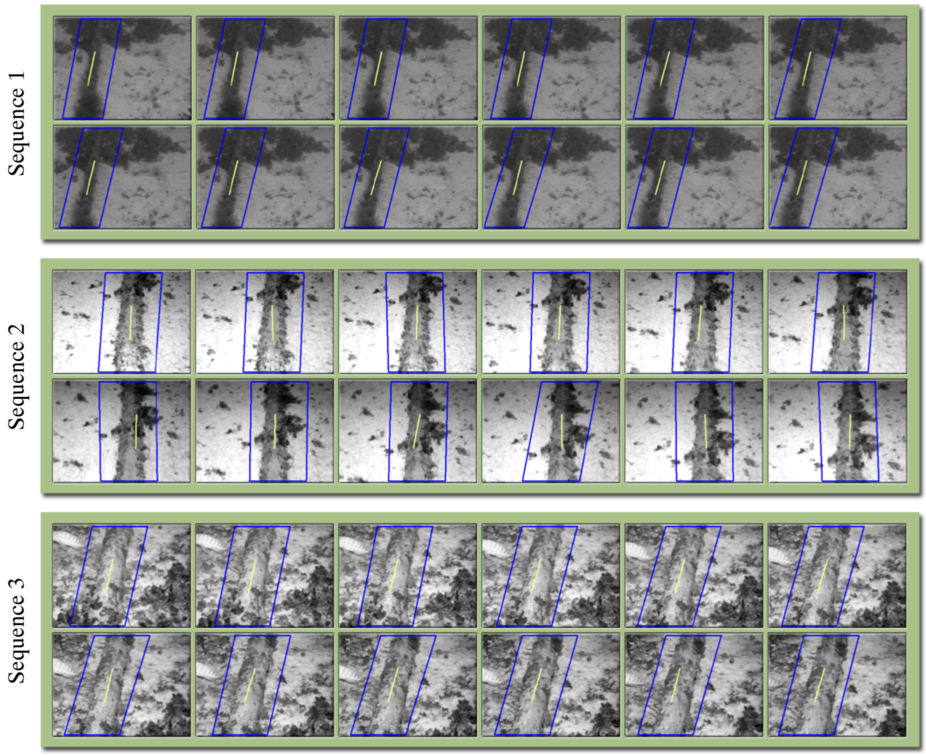


Fig. 7. Vision system output for an excerpt of three different sequences with the ROI superimposed in blue. The yellow line represents the computed main cable axis and thus a possible command to the AUV navigation controller

been released from the support ship, goes down until a certain distance to the seabed is reached. The second and third stages comprise, respectively, searching for the cable in a predefined exploration area and tracking it once found. Finally, the vehicle returns to the starting point after having achieved the limits of the exploration area while tracking the cable.

Three different kinds of sensors are used to suitably carry out such mission: various sonars, a compass and a camera. Furthermore, the position of the vehicle is estimated by means of an acoustic positioning system of the so-called Long Base Line (LBL) type.

3.2 Behaviour Description

Taking into account the aforementioned general way of acting for the AUV, the reactive control layer of the vehicle was split into six primitive behaviours. Some of them appear in the classical literature about behavioural architectures, but others are specific of this application. They all are described in the following:

- *Stay on region* prevents the AUV from straying from the area to be explored. The behaviour is exclusively activated when the vehicle is close to the limits of the exploration area. In such a case, a vector that moves the vehicle away from those limits is generated, being its magnitude directly related to the corresponding distance: the closer to the limits, the larger the magnitude.
- *Avoid obstacles* allows the vehicle to avoid navigational barriers such as rocks, algae or, even, other possible cooperating vehicles. In this case, a vector in the opposite direction to the obstacles is generated. The magnitude of the vector is again variable, now according to the distance that separates the AUV from the obstacles ahead.
- *Avoiding the past* tries to avoid the well-known trapping problem characteristic of reactive strategies (Balch and Arkin, 1993). For such a purpose, a local map of the most recent AUV's path is used. When the vehicle is detected in essentially the same area for a long time, this behaviour becomes active generating a vector whose direction favours the exploration of new regions of the environment. In this case, the magnitude of the vector is proportional to the size of the area where the vehicle has been trapped into.
- *Cable detection and tracking* moves the vehicle strategically through the exploration area in search of a sufficient evidence of the presence of the cable. Specifically, after having acquired the working

depth through a vertical path from the surface, the AUV executes the sweeping stage performing a zigzag movement on the exploration area until the cable is found. Although other more optimised strategies could have been devised, it is important to note that it has been assumed a total lack of information about the location of the cable, so that there are no many more alternatives but an exhaustive or near-exhaustive search.

Once the cable has been detected, the tracking stage starts. At this point, the AUV can be oriented in any one of the two possible –and opposite– directions to start tracking the cable. The particular choice is based on a predefined parameter which establishes a certain range of preferred orientations.

Along the tracking, two different tasks are sequentially executed: the first one tries to keep the cable oriented vertically in the field of view (FOV), while the second task intends to maintain the cable in the central area of the FOV. In this way, improvements in both the cable visual detection and the longitude and smoothness of the vehicle's path are expected.

As can be anticipated for a real application, anomalous situations can arise. In particular, the cable can disappear from the images because the AUV's course has drifted apart from the actual cable location. In such cases, a suitable recovery mechanism is activated, consisting in making the behaviour return to its internal search state, where the vehicle acquires the aforementioned zigzag movement. However, now the area to be explored is reduced using the vehicle's trajectory during the past tracking stage. This trajectory is fitted by a straight line and a new search zone is determined computing the intersection between such line and the limits of the exploration area (see figure 8). Note that the dimensions of that zone can be readjusted according to the AUV's manoeuvrability.

- *Keep distance to seabed* tries to maintain the distance to the seabed constant in order to keep the apparent width of the cable in the images also constant. In this way, the vision subsystem can assume that the separation between both sides of the cable does not vary, and use this information to reduce its probability of failure.
- *Go home*, finally, makes the vehicle go to the starting point of the mission. Two different steps are carried out: first, the AUV approaches that point keeping a certain distance to the seabed; afterwards, it goes up until the sea surface is reached. In both cases, the magnitude of

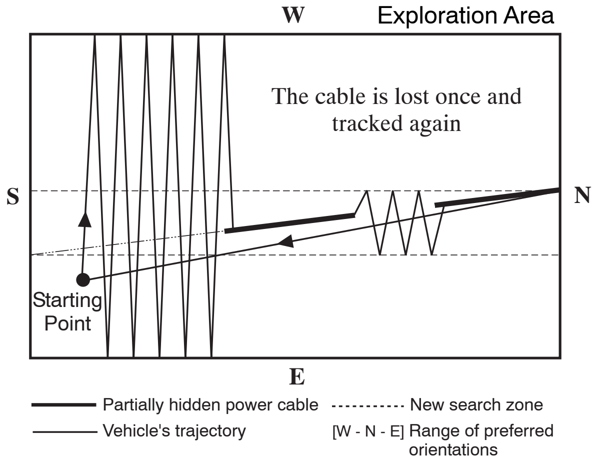


Fig. 8. AUV's trajectory for a typical mission

the output vector is proportional to the proximity to the intermediate/final goals considered.

Figure 9 summarises the way how the aforementioned components of the control architecture are organized. As can be seen, a supervisor has also been added. From a functional point of view, this component simply turns behaviours on and off according to the mission stage where the vehicle is, in order to avoid conflicts among them.

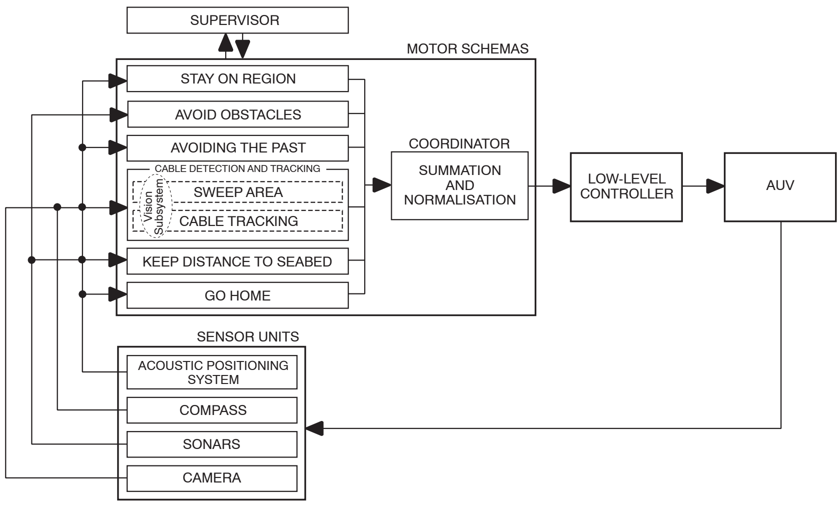


Fig. 9. Main components of the control system

4 Some Simulated Results

To illustrate the behaviour of the proposed control architecture inside the simulation environment $NEMO_{CAT}$, some graphical results for three different missions are reported in the following. These results were obtained by using the dynamic model of the underwater vehicle GARBI (Ridao *et al.*, 2001) and applying the behaviour gains[1] shown in table 2. As will be seen, the vehicle suitably located and followed the cable/pipeline in all cases.

Table 2. Behaviour gain values

Behaviour	Gain value
Stay on Region	2.00
Avoid Obstacles	6.25
Avoiding the Past	2.35
Cable Detection and Tracking	1.75
Keep Distance to Seabed	2.75
Go Home	1.50

In the first mission, the AUV, after the diving stage, was trapped into a box-shaped canyon. Later, as was to be expected, the activation of the "avoiding the past" behaviour allowed the vehicle to escape from that undesirable situation. The characteristic zigzag movement of the sweeping stage was then resumed. When tracking, the cable was lost on one occasion and subsequently tracked again after a restricted search process carried out on a small region of the exploration area. The mission finished with the return of the vehicle to the starting point. All the results for the mission are given in figure 10 where (a) details the AUV's trajectory previously described, (b) represents the activity of the different architecture's behaviours along the mission, (c) shows how the vehicle's trajectory is adapted to the unevenness of the seabed, and, finally, (d) provides information about the cable position and orientation in the images captured by the camera attached to the vehicle. In this last case, two plots are displayed corresponding each to the two tracking stage executions. In the plots, the y-axis is a percentage that reaches its maximum value when the orientation of the cable is vertical and it is located in the middle of the image.

[1] Gains encode the relative strength of each behaviour

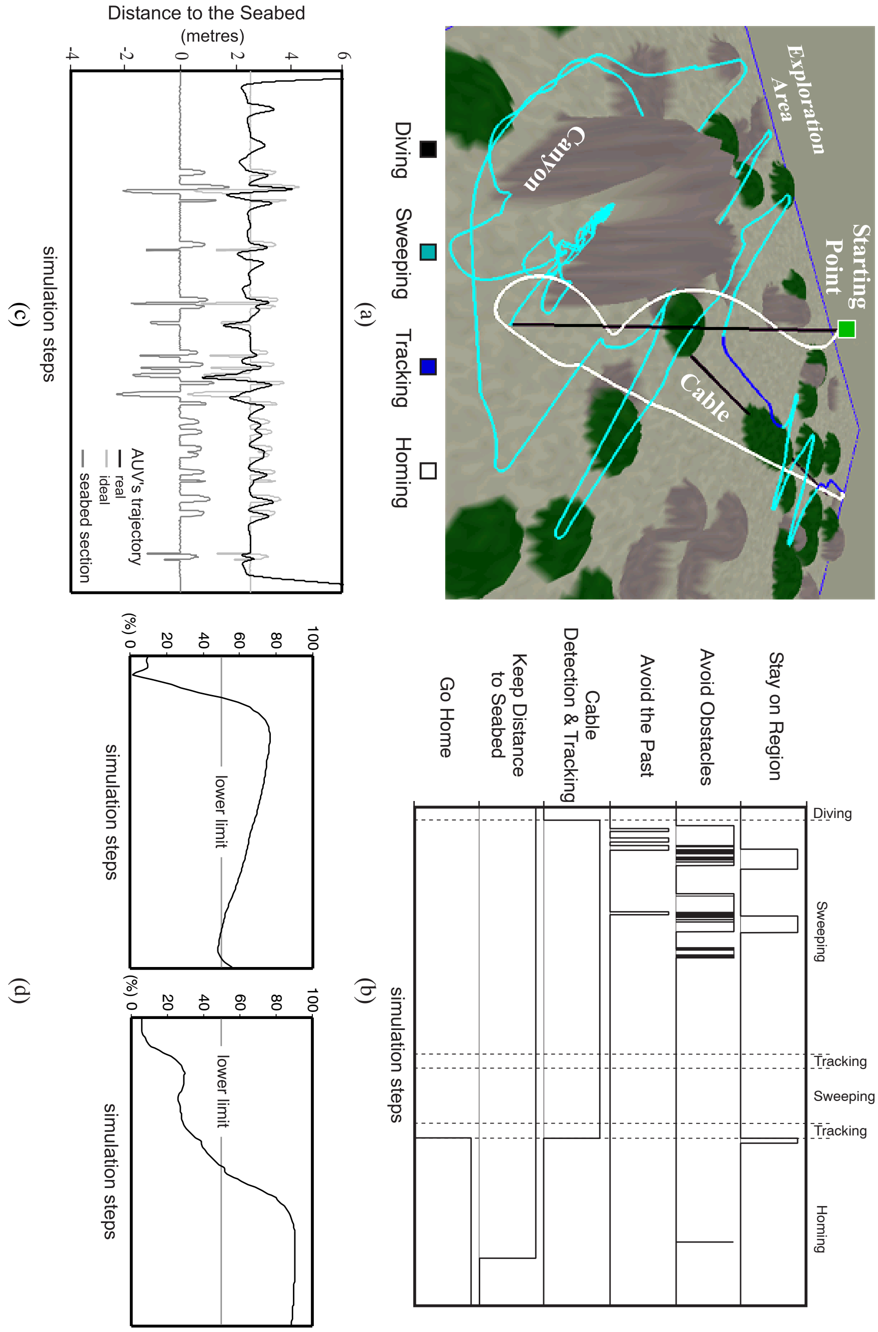


Fig. 10. Results for the first simulated mission

From a complexity point of view, the first and the second missions were very similar. In the latter, however, the curvature of the cable was significant. The reader should notice that the restricted search process which is activated when the cable/pipeline is lost assumes a certain rigidity for that object. Despite this fact, as can be observed in figure 11, the process continued being effective.

Finally, the third mission was more realistic than the previous ones. In this case, the seabed of the simulated underwater environment was represented by a finite plane mapped with a special texture obtained by manually applying a mosaicking process to an excerpt of an image sequence of a real cable used to test the vision system. Figure 12 shows in detail the results for this mission.

5 Conclusions

A behavioural control architecture for locating and tracking an underwater cable/pipeline on the basis of a vision subsystem has been presented. The whole system has been validated using a 3D OpenGL-based simulation environment named $NEMO_{CAT}$. Missions of different complexity were successfully carried out on that simulator. Detailed results for the most representative missions have been shown.

The proposal, nevertheless, is not without limitations. More precisely, undesirable trapping situations may arise due to the local minima problem (Koren and Borenstein, 1991). Figure 13 shows, by way of example, how the underwater cable tracker is unable to escape from a deep U-shaped canyon in spite of using the behaviour *avoiding the past.* The main cause of this fact is found in the locality of the approach which does not allow the robot to always take the right action due to the lack of information about the navigation environment. Another chapter of this book will address the problem.

Acknowledgments

This study has been partially supported by project CICYTDPI2005-09001-C03-02 and FEDER funds. Its authors wish to thank GESA for supplying them the cable image sequences used to test the vision system and also to the members of the Computer Vision and Robotics research group of the University of Girona (Spain) for providing them with the hydrodynamic model of their underwater vehicle GARBI.

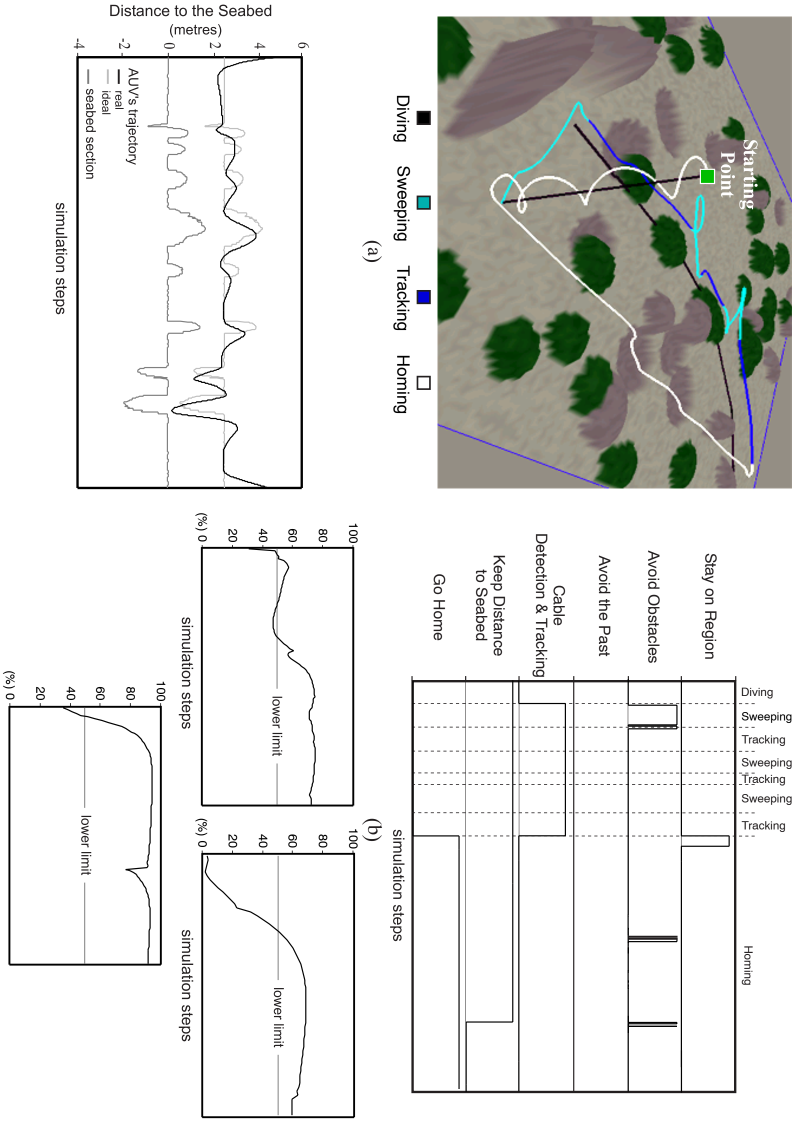


Fig. 11. Results for the second simulated mission

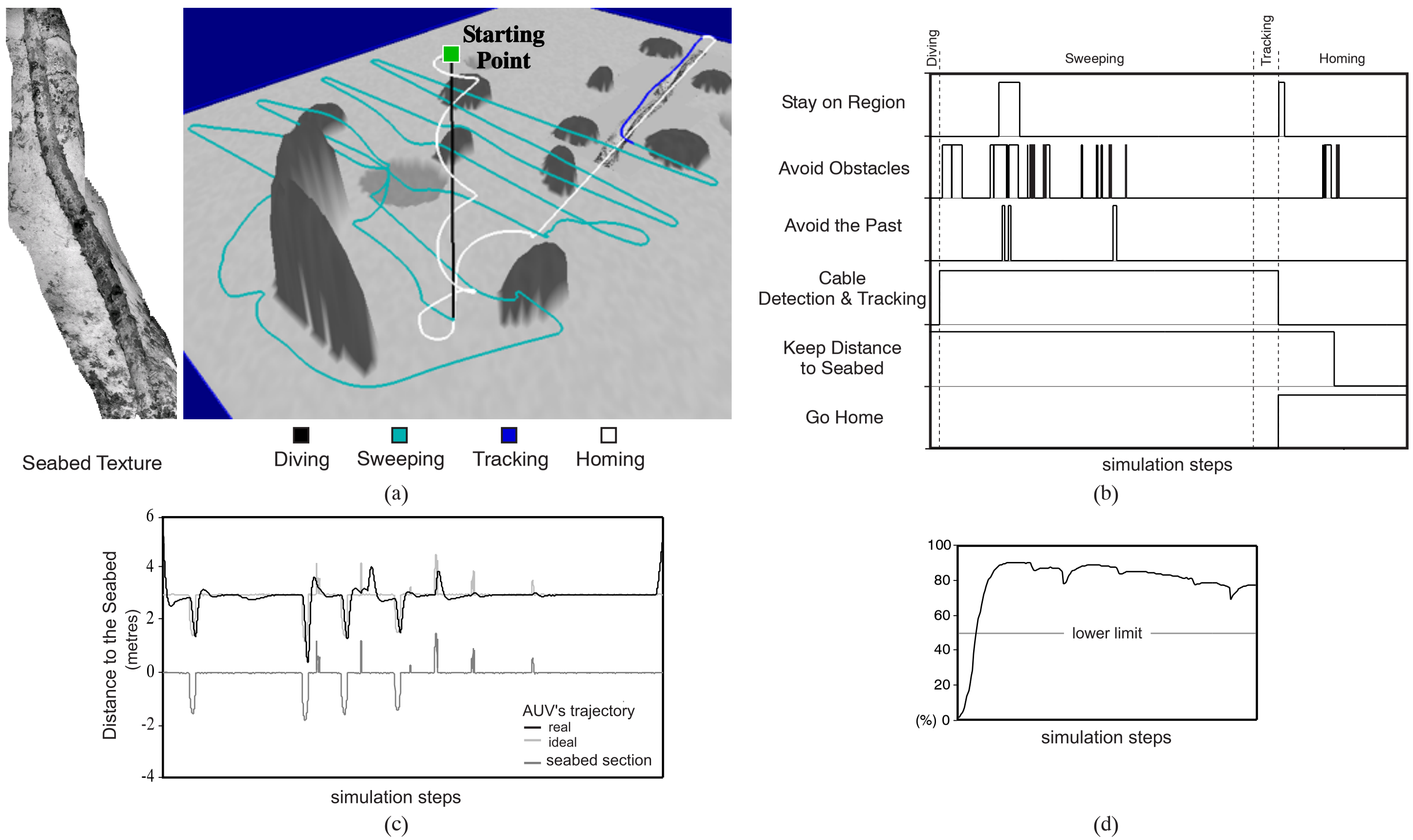


Fig. 12. Results for the third simulated mission

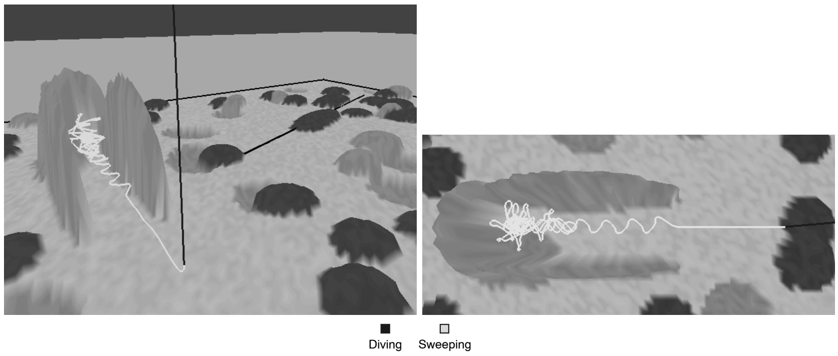


Fig. 13. The AUV, after the diving stage, is trapped into a deep box-shaped canyon

References

Antich, J. 2006. Reactive Robotics: A Paradigm Not Limited to Simple Tasks. Master's thesis, University of the Balearic Islands.

Antich, J. and Ortiz, A. 2004. An underwater simulation environment for testing autonomous robot control architectures. In Proceedings of the IFAC Conference on Control Applications in Marine Systems, pages 509–514.

Arkin, R. 1989. Motor schema-based mobile robot navigation. Intl. Journal of Robotics Research, **8**(4): 92–112.

Balch, T. and Arkin, R. 1993. Avoiding the past: A simple but effective strategy for reactive navigation. In Proceedings of the Intl. Conference on Robotics and Automation, pages 678–685.

Duda, R. and Hart, P. 1973. Pattern Classification and Scene Analysis. John Wiley & Sons.

Khatib, O. 1986. Real-time obstacle avoidance for manipulators and mobile robots. Intl. Journal of Robotics Research, **5**(1): 90–98.

Koren, Y. and Borenstein, J. 1991. Potential field methods and their inherent limitations for mobile robot navigation. In Proceedings of the Intl. Conference on Robotics and Automation, pages 1398–1404.

Ortiz, A., Simó, M., and Oliver, G. 2002. A vision system for an underwater cable tracker. Intl. Journal of Machine Vision and Applications, **13**(3): 129–140.

Ridao, P., Batlle, J., and Carreras, M. 2001. Dynamics model of an underwater robotic vehicle. Technical Report IIiA 01-05-RR, University of Girona.

CHAPTER 7

Traversability and *Tenacity*: two new concepts improving the navigation capabilities of reactive control systems

J. ANTICH and A. ORTIZ

Department of Mathematics and Computer Science
University of the Balearic Islands
Cra. Valldemossa, km 7.5 - 07122 Palma de Mallorca (Spain)
{javi.antich,alberto.ortiz}@uib.es

Reactive mobile robot navigation based on potential field methods has shown to be a good solution for dealing with unknown and dynamic scenarios, where timely responses are required. Unfortunately, the complexity of the tasks which can successfully be carried out is limited by the inherent shortcomings of the approach such as trapping situations due to local minima, difficulties passing among closely spaced obstacles, oscillations in narrow corridors, etc... This chapter outlines a set of strategies which overcome the first of the aforementioned limitations by computing an adaptive navigation function on the basis of such artificial potential fields. As a result, navigation can be achieved in very difficult underwater (and land) scenarios where obstacles adopt maze-like configurations. A comparative study on the path length performance of our proposal with regard to other algorithms from the related literature is also presented.

1 Introduction

In robotic navigation, potential field methods (PFM) (Khatib, 1986) are a well-known solution for dealing with unknown and dynamic scenarios such as the submarine by taking into account the reality of the environment during the robot motion. The characteristic elegance and simplicity of the approach when representing and successfully solving a path-planning problem in real-time explains its extensive application in this field. However, substantial shortcomings have been identified as problems inherent to this principle (Koren and Borenstein, 1991). *Getting stuck in local minima* is the best-known and most often-cited problem with PFMs. As a result, several obstacle configurations such as the typical U-shaped canyon may lead to undesirable trapping situations. Another drawback associated with this kind of systems is related to the lack of an oscillation-free motion when the robot navigates among very close obstacles at a high speed. Finally, the impossibility to go through small openings constitutes the last significant problem of PFMs.

In this context, the contribution of this work is twofold:

- On the one hand, a solution to the local minima problem is given according to the so-called concepts of *Traversability* and *Tenacity* —or T^2, in brief. As a result, navigation is achieved in very difficult scenarios, even including maze-like environments. Unfortunately, the completion of the mission cannot be always guaranteed due to the appearance of cyclic-oscillatory behaviours.
- On the other hand, three different algorithms, based on the T^2 principles, are also put forward to ensure, whenever possible, the attainment of any navigation task.

The rest of the chapter is organized as follows: section 2 introduces the basic framework under which the new family of navigation strategies, generically called T^2, will be defined, while sections 3 and 4 provide a detailed description of it; a comparative study on the path length performance of the proposal is presented in section 5; and, finally, some conclusions are given in section 6.

2 Framework

The classic potential field approach proposed in (Khatib, 1986) constitutes the basic framework for the application of the novel T^2 family of navigation strategies. It computes the motion of the robot on the basis of two simple behaviours: *GoTo* and *AvoidObstacles*. More precisely, the former generates an attractive force in direction to the goal, while the latter considers obstacles as repulsive surfaces. The robot follows the negative gradient of the resulting potential field towards its minimum, whose position is expected to coincide with the goal point.

3 Fundamentals of the T^2 Navigational Approach

The inability to move the robot away from the goal direction in a non-momentary and strategic way is the main cause of the undesirable trapping situations suffered by the reactive control paradigm and, in particular, by the artificial potential fields. Fig. 1(a) shows an example where a robot adopting this last approach is unable to escape from a U-shaped obstacle. In the following, a solution will be given by applying two concepts, *Traversability* and *Tenacity* —or T^2, in brief—, in the context of the so-called navigation filter. As a result, those trapping situations linked to the local minima problem will be successfully avoided. However, it will be also seen that this module does not always ensure the completion of the mission. In the next section, several changes will be carried out in order to guarantee the convergence. Finally, the reader should notice that the navigation filter is not an isolated control module but a new component of a generic behaviour-based control system. Fig. 1(b) illustrates its integration into the classic potential field approach. The resulting control diagram will be taken as a reference from now on.

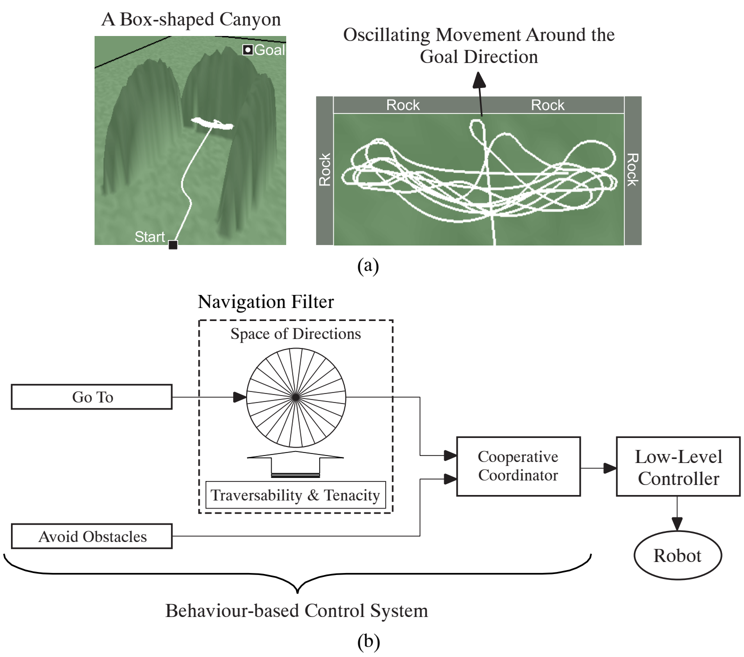


Fig. 1. (a) A typical trapping situation for reactive control systems; (b) integration of the navigation filter into the classic potential field approach.

3.1 The Navigation Filter

The main task of the navigation filter is the appropriate alteration of the direction of the motion vector generated by the *GoTo* behaviour in order to rapidly overcome any obstacle irrespective of its size and shape (see fig. 1(b) again). Such change is carried out according to the *Traversability* and *Tenacity* principles. In short, the first one suggests, on the one hand, banning those directions where an obstacle has been detected and, secondly, choosing an obstacle-free direction close to the desired one — *GoTo*'s response— when it has been banned. In this last case, the *tenacity* principle determines, finally, what obstacle-free direction will be selected among all the available alternatives. Next, both principles will be treated in depth.

3.1.1 The Traversability Principle

The application of the *traversability* principle requires the division of the space of directions around the robot into K identical angular regions as it is shown in fig. 2(a). These regions can be additionally classified as *banned* or *allowed*. Specifically, a region will belong to the former group when at least one obstacle is known to be in the range of directions which consists of. The non-existence of obstacles, on the other hand, characterises the latter.

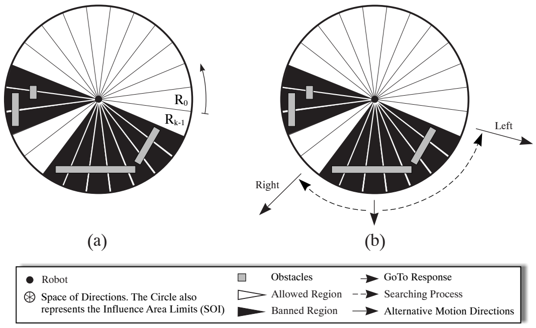


Fig. 2. Exemplifying the implementation of the *traversability* principle: (a) division of the space of directions into K regions, labeling them as *allowed* or *banned*; (b) selection of two obstacle-free motion directions.

Based on the previous information, this principle is intended to forbid the robot's movement in directions where the presence of obstacles has recently been determined, avoiding thus unnecessary and unsuccessful displacements in the task of looking for a path towards the goal point. With this purpose, after receiving the response of the *GoTo* behaviour, its viability is studied according to the above-mentioned premise. Changes are required only if the direction of such motion vector lies in a banned region. Otherwise, the navigation filter will not act, generating as output the same input vector. In such a case, note that the classic potential field approach is really used to produce the response of the control system. Assuming the first situation which has been described, two alternative directions, generically labeled as left and right, will be obtained as a result of a double searching process, clockwise and counterclockwise, for the first allowed region starting from the desired direction of motion (see fig. 2(b) for an example). The final decision about choosing one direction or another depends on the *tenacity* principle.

3.1.2 The Tenacity Principle

Taking fig. 1(a) as an example, the trajectory of a robot which has been trapped can be seen, in a simplified way, as the result of two simple movements: forward and backward. These movements alternate the control of the robot each time the vehicle moves excessively away from the goal direction. The *tenacity* principle precisely tries to give a solution to that oscillating/hesitant behaviour by preventing the robot from drastically changing its direction of motion. In this way, progress will be always ensured removing thus the main cause of any trapping situation. Regarding the implementation details, remember that two alternative motion directions labeled as left and right are obtained when the response of the *GoTo* behaviour lies in a banned region. Under these circumstances, one of such directions has to be selected as output of the navigation filter. With this aim, the *tenacity* principle is applied by choosing left or right in coincidence with the last decision made. Finally, note that, despite the obvious simplicity of the concept, this principle has proven fully effective.

3.1.3 Need for Remembering the Obstacles

Purely reactive systems such as the classic potential field approach react directly to the world as it is sensed, avoiding the need for intervening any kind of abstract representational knowledge. The sentence “what you see is what you get” faithfully summarises this idea. However, the local information provided by the robot’s sensory equipment may not be enough to solve a given navigation task. Our navigation filter, against this problem, keeps and uses information regarding the obstacles beyond the robot's immediate sensory range. The approach is, nevertheless, still reactive since no path planning is ever conducted on such information. By way of example, fig. 3 shows how the navigation filter is able to remember the presence of obstacles in directions where, currently, an obstacle-free space is being locally detected. To this end, the approximate location of the obstacles in the environment is memorised. It is important to note, however, that the character of these data is temporary, being removed when the robot is sure that the corresponding obstacle has been successfully overcome. Further details about this subject will be given later.

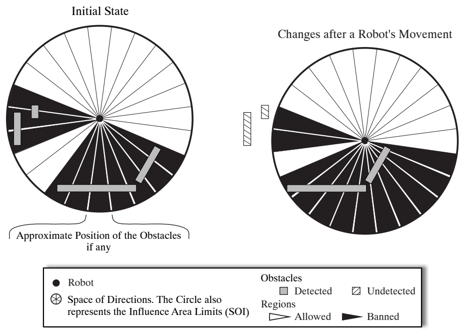


Fig. 3. Memorising the obstacle locations while navigating.

3.2 The Emergent Global Behaviour

The combined application of both the *traversability* and *tenacity* principles results in an emergent global behaviour which can be summarised in the next three points (see fig. 4 for a simple example):

1. When the robot is navigating far from obstacles, it heads for the goal following a straight path. During this period of time, the navigation filter remains inactive generating as output the same input motion vector.
2. After the detection of an obstacle, the robot follows its contour in a certain direction. To this end, changes are required on the response of the *GoTo* behaviour based on T^2. Of special interest is, however, the first time that a change has to be performed. In such a case, the *tenacity* principle cannot be applied due to the lack of a previous

decision, so that an additional selection criterion is needed for these particular situations. This first decision will determine the direction taken by the robot when following the obstacle boundary. In short, the decision is made on the basis of how the previous obstacle was followed, or lacking this information —first obstacle—, according to a minimum turn criterion.

3. Finally, the robot knows that the obstacle has been overcome when the direction to the goal becomes free of obstacles, that is, not banned. At that moment, the filter is reset losing thus all the previously kept information —obstacle positions—, which is no longer necessary for the navigation task.

These stages will be sequentially executed in the order specified so many times as obstacles the robot finds on its way towards the goal point. To finish, note that the *AvoidObstacles* behaviour also plays an important role on the final robot's behaviour. More precisely, it helps to maintain a reasonable distance between the robot and the contour of an obstacle when it is being followed.

3.3 Major Shortcoming

The successful mission completion cannot be always guaranteed by means of, only, the navigation filter —that is, the T^2 principles. Fig. 5 shows, precisely, an example confirming this fact. As can be observed, the robot was not able to reach the goal point by generating a cyclic behaviour around a G-shaped obstacle. In the next section, three different solutions to this problem will be given. They slightly modify the way how the robot decides to leave and follow, in one direction or another, the boundary of an obstacle.

4 Three T^2-based Algorithms with Provable Guarantees

This section constitutes a continuation of the previous one, where a new component named navigation filter was incorporated into a control system based on the potential fields approach to avoid the known and, up till then, unsolved local minima problem. In this sense, at present, it is intended to step forward by ensuring, whenever possible, the achievement of the goal point for any mission. To this end, three different algorithms will be proposed which alter the way how such navigation filter takes two kinds of decisions related to: on the one hand, the direction for following

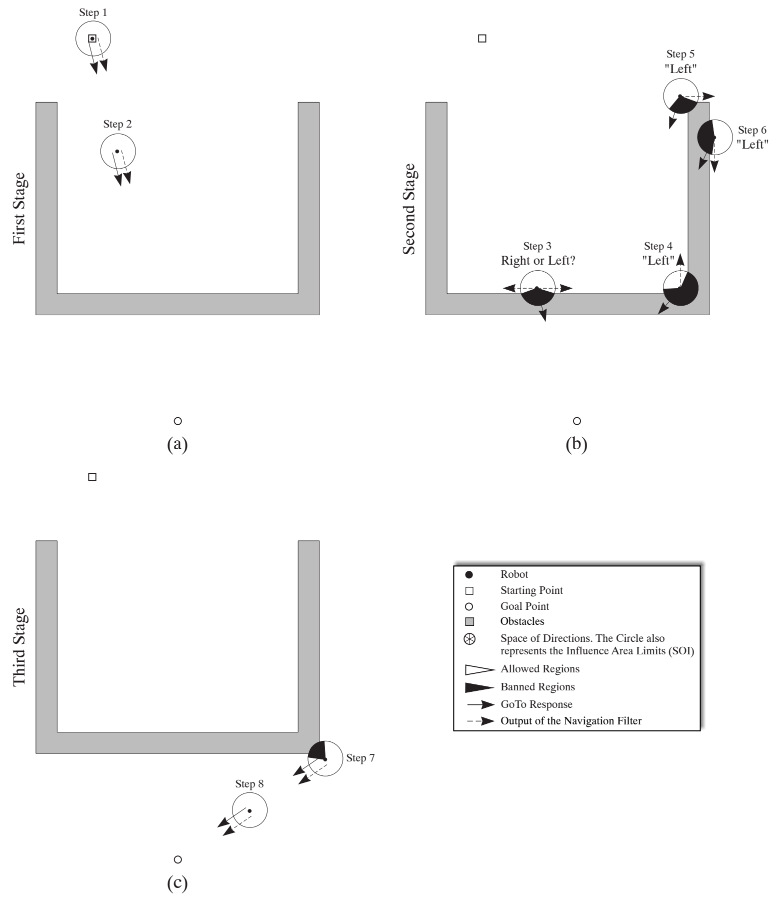


Fig. 4. Escaping from a U-shaped obstacle: (a) direct path to the goal point; (b) following the contour of the obstacle to the robot's left (note that this direction is defined in step 3 by selecting the option labeled as left based on a minimum turn criterion); (c) reset of the navigation filter returning, afterwards, to the same situation as in (a) from which the goal is finally reached.

the contour of an obstacle and, on the other hand, the leaving of such contour for trying, afterwards, to definitely reach the goal through a straight path (see fig. 6 for a graphic illustration of both types of decisions). These algorithms are specifically called *Random* T^2, *Connectivity* T^2 and *Bug-based* T^2, representing all of them the new family of algorithms T^2. Next, each of the members of such family will be briefly described (the reader is referred to (Antich, 2006) for further details).

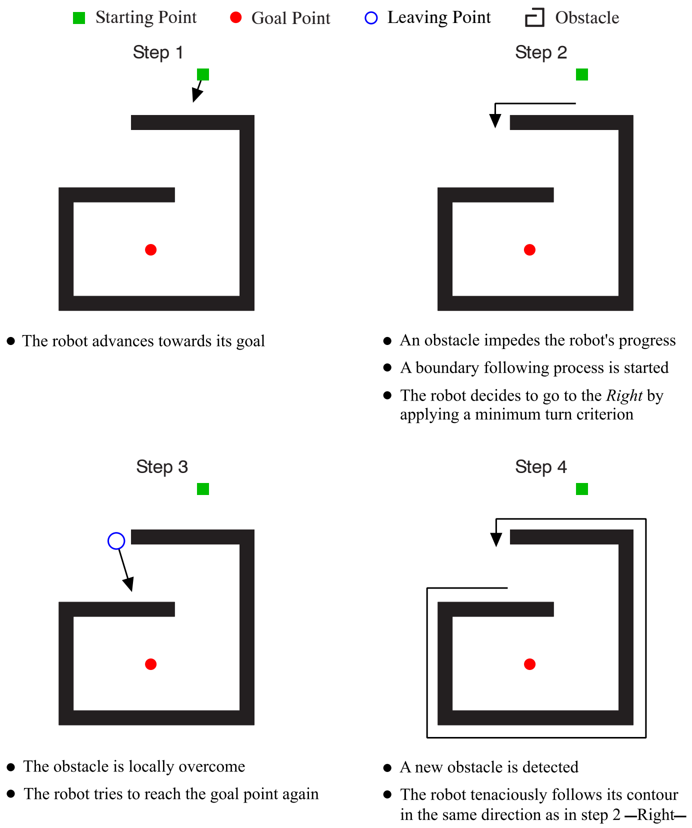


Fig. 5. A mission not solved by the navigation filter. Note that after step 4, the third step will be executed again, generating thus a cyclic behaviour. The robot's trajectory has been computed by hand according to the known three stages of the strategy (refer to section 3.2).

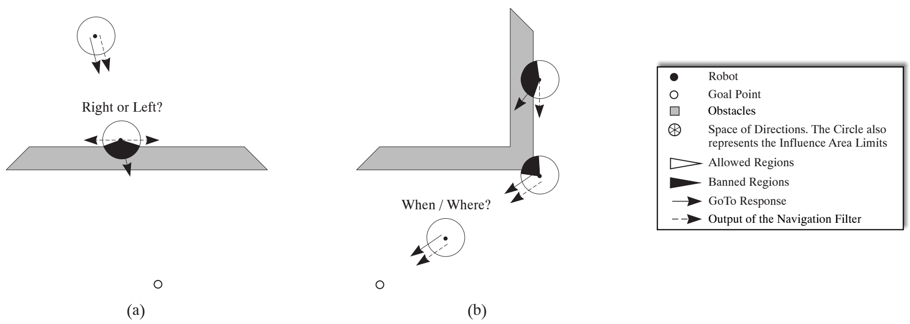


Fig. 6. Decisions having influence on the convergence —goal achievement guarantee— of an algorithm based on the T^2 principles: (a) selection of the contour following direction of the obstacles; (b) leaving the obstacle boundaries.

4.1 The Strategy *Random* T^2

Random T^2 is the simplest strategy of the family of algorithms being proposed. In this sense, non-extra data and reasoning are needed apart from the ones associated with the navigation filter to make the decisions pointed out in fig. 6. More precisely, these are the specific criteria applied by the strategy regarding such decisions:

- In the context of T^2, the completion of a mission may be fully conditional on the following of the contour of some obstacles in certain directions —left or right— depending on the particular features of the navigation environment. These features are unknown by the robot in accordance with its reactive nature, concluding thus that it will never be possible to properly define a fixed beforehand criterion for choosing the contour following direction which is valid for any mission without exception. Note that it was precisely the mistake made by the basic T^2 approach presented in section 3 where the same direction was always taken. *Random* T^2, keeping the previous considerations in mind, adopts the simplest —and also less efficient— solution to the problem by selecting that direction in a random way. In this manner, if an obstacle is followed in the wrong direction, the robot will be able to unconsciously/probabilistically rectify its decision in subsequent occasions reaching, finally, the goal point.
- On the other hand, as for the leaving of the contour of the obstacles, no changes have been performed with regard to the original T^2 approach. In this way, the boundary of an obstacle will be essentially left when the direction to the goal point becomes *allowed* —free of obstacles. Note that, from a geometric point of view, such circumstance exclusively arises in situations where the robot faces the goal while following the obstacle contour.

Finally, by way of example, fig. 7 shows how the strategy would suitably accomplish the mission presented in fig. 5 which was not solved by the navigation filter alone.

4.2 The Strategy *Connectivity* T^2

The strategy *Connectivity* T^2 is intended to improve the efficiency of the previous algorithm in the search of a path towards the goal point avoiding that, fundamentally, parts of the environment —obstacle boundaries— which have already been explored by the robot in an unsuccessful way

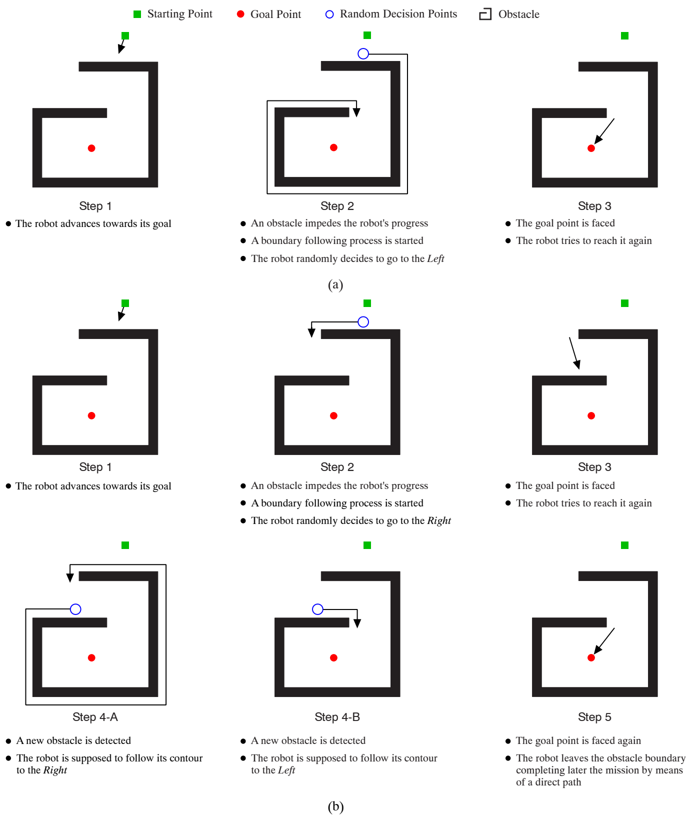


Fig. 7. Exemplifying the behaviour of the strategy *Random* T^2 in a scenario with a G-shaped obstacle. (a) and (b) are the two paths which result from considering all the possible decisions of the algorithm with regard to the selection of the obstacle contour following directions. The situation shown in (b) requires, however, an additional comment. Either step 4-A or step 4-B will be randomly chosen to be carried out by the robot. In the former case, step 3 will be afterwards executed again. In the latter, on the contrary, the control flow will go to step 5 where the goal point will be finally achieved.

are covered again (remember fig. 7 where step 4-A might be executed more than once). To this end, the concept of *key area* is introduced into the original T^2 approach. Specifically, it is essentially an artificial landmark virtually located next to an obstacle that indicates in what directions —left and/or right— the contour of such obstacle has been followed starting from that precise position. Fig. 8 illustrates the mentioned concept in a simple mission. As can be observed, each time the robot detects a new obstacle and, in consequence, has to decide to follow its contour in one direction or another, a key area is created for registering the decision taken. Note that this information, in more complex scenarios such as that of fig. 7, will allow the robot not to repeat decisions on the same obstacle favouring thus an extensive and fast exploration of the navigation environment. On the other hand, as for the leaving of the obstacle boundaries, just like it happened in the algorithm *Random* T^2, no changes have been carried out on the proposal of section 3.

Finally, fig. 9 shows an example of the application of the strategy.

4.3 The Strategy *Bug-based* T^2

Based on the T^2 principles, the last strategy which is proposed is called *Bug-based* T^2. Specifically, this strategy ensures convergence —that is, the mission completion whenever possible— by slightly modifying the way how the navigation filter decided to leave the contour of the obstacles.

As was already explained in section 3.1, the navigation filter makes the robot abandon the boundary of an obstacle when, essentially, the direction to the goal becomes *allowed* —free of obstacles. It is important to note that, from a geometric point of view, such circumstance exclusively arises in situations where the robot faces the goal point while following the obstacle contour. These points of leaving, taking into account the local information which is really managed by the strategy, correspond to the most promising and closest places from where the robot would be supposedly capable of completing the mission through a direct path. This advantage, however, contrasts with the lack of convergence which is derived from such decisions (remember fig. 5).

Regarding the leaving of the contour following process, other interesting criteria can be found in the related literature. More precisely, the strategy *Bug2* (Lumelsky and Stepanov, 1987) suggests that the robot stops following the boundary of an obstacle when its trajectory cuts the straight segment joining the starting and the target points, also named Main Line —or M-line, in brief. As a result, this strategy guarantees the goal achievement at the expense of, generally, longer paths. Fig. 10(a) and (b) compare the conditions for leaving applied by respectively the T^2 and *Bug2* approaches, from the viewpoint of the resultant robot's trajectory in a simple scenario.

Both advantages, a good path length performance as well as convergence, can be really attained by combining and slightly altering the previous conditions in a proper way. On the one hand, the leaving conditions for T^2 (C1) and *Bug2* (C2) are defined as follows:

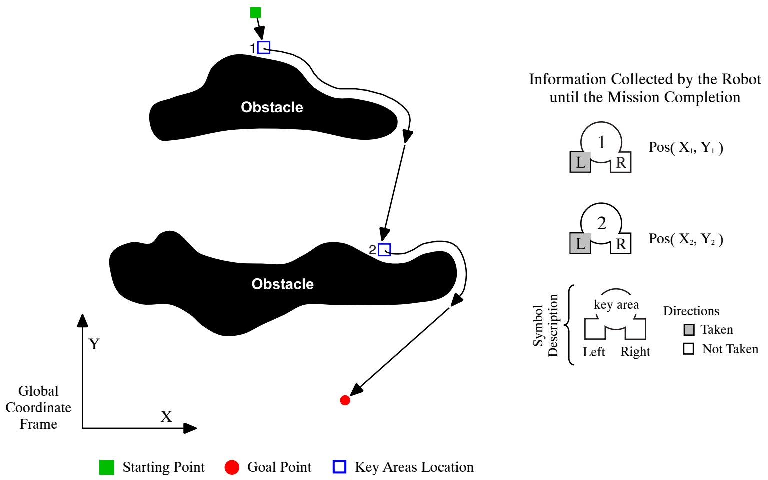


Fig. 8. The *key area* concept.

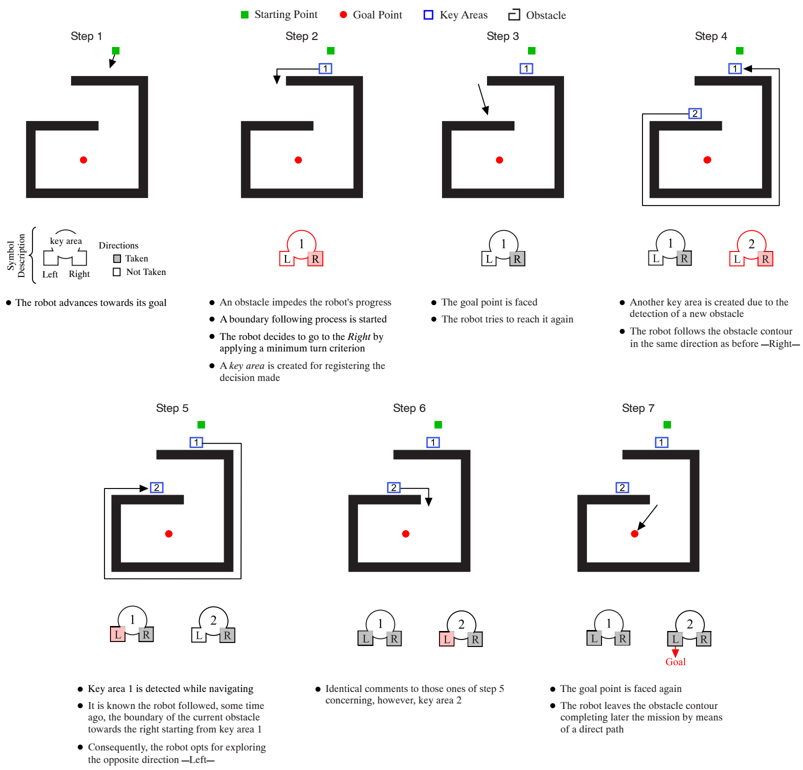


Fig. 9. The strategy *Connectivity* T^2 solving a mission with a G-shaped obstacle. Note that the key area information kept by the robot is also represented in the figure for each step being highlighted in red the latest changes.

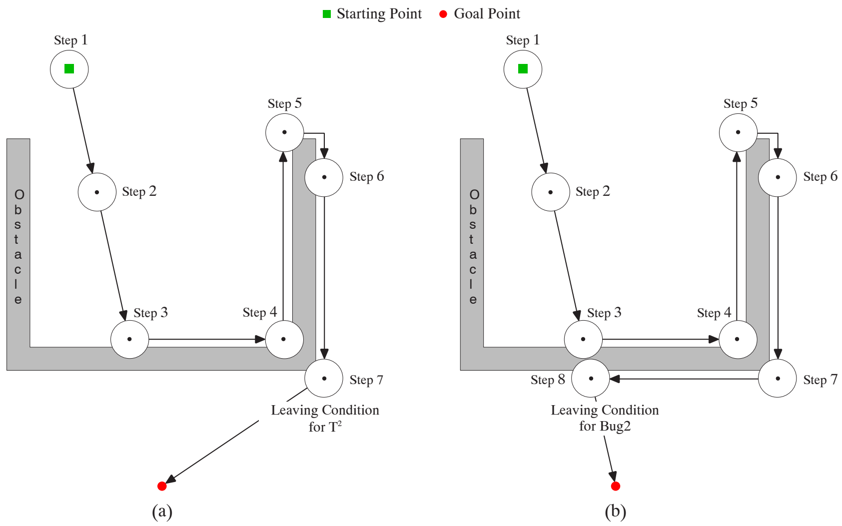


Fig. 10. Comparing the leaving conditions linked to the T^2 (a) and *Bug2* (b) strategies.

C1) The navigation filter indicates there is a free-obstacle path towards the goal point. Moreover, this must be the first time the robot leaves the obstacle at approximately that same position.

C2) The robot's trajectory cuts the M-line. Additionally, the distance from the robot to the goal has to be shorter than the one associated with the last time this condition was satisfied. Initially, this distance is initialised to the M-line's length.

On the other hand, either C1 or C2 must be met for the leaving to occur. At this point, notice that, contrary to the *Bug2* algorithm, the M-line concept is not static for *Bug-based* T^2. More precisely, each time the leaving condition C1 is satisfied, the M-line is modified by considering as new starting point the current robot position[1]. This fact together with the limited number of times that such condition can be fulfilled[2] allow proving the convergence of *Bug-based* T^2. The formal proof can be found in (Antich, 2006) together with some simulated and real experiments.

[1] In such a case, the distance used by condition C2 is also reinitialised

[2] This assertion is based on the assumption that there is a finite number of obstacles in the navigation environment

5 A Comparative Study

In the following, a comparative study on the performance of the new set of navigation strategies —*Random* T^2, *Connectivity* T^2 and *Bug-based* T^2 to be precise—, generically called T^2, will be presented and discussed in detail. Four algorithms from the related literature have been considered. Among them, there are some popular approaches while the others have been recently published. On the other hand, the comparison is carried out from the point of view of a single criterion which is the length of the path generated between the starting and the goal points.

5.1 Strategies Considered

Several strategies have been selected to take part in the comparative study against the three proposed navigation algorithms. Their main features are summarised next:

- *Avoiding the Past* (Balch and Arkin, 1993). The robot moves to the user-specified goal point while being repelled from locations which were already visited. With this purpose, a local map of the environment implemented as a two-dimensional grid is stored in memory, where a different value is assigned to visited and non-visited locations. As the robot visits an area more times, the values of the corresponding cells in the grid increase and, consequently, the resultant repulsive force exerted by such cells increases as well. In this way, it is intended to favour the continuous exploration of new regions of the navigation environment avoiding thus, at least apparently, the robot gets stuck into a local minimum.
- *Learning Momentum* (LM) (Lee and Arkin, 2001). This strategy adjusts the behavioural parameters of a particular reactive control system at runtime instead of using static values. A module called *Adjuster* is responsible for this task. This module, based on recent experience and a set of heuristic rules, identifies when good progress to the goal is being made. According to this, the gains as well as other parameters of the three behaviours making up the control system —GoTo, AvoidObstacles and Noise— are properly altered.
- *Micronavigation* (μNAV) (Scalzo *et al.*, 2003). This approach tries to solve the problem of mobile robot navigation from a minimalist point of view by only using, as its author says, a handful of bytes. Specifically, the robot is provided with a hierarchy of simple behaviours

designed for smooth obstacle avoidance through the *equipotential line* concept and for escaping from concavities.

- *Bug2* (Lumelsky and Stepanov, 1987). This is a representative member of one of the most popular families of algorithms for path planning with incomplete information named *Bug*. The goal achievement guarantee whenever possible can be found among its main features. As for the strategy, two basic behaviours, GoTo and ContourFollowing, alternate the control of the robot. Initially, the GoTo behaviour is active. The detection of an obstacle, on the other hand, starts the contour following process. Such process is left by the robot when it cuts the virtual line connecting the starting and the goal points, also called *Main Line*. Finally, notice that some of the concepts introduced by this algorithm were suitably incorporated into our approach *Bug-based* T^2 to ensure its convergence (see section 4.3). This fact explains the inclusion of this strategy in the comparative study.

5.2 Results for a Representative Set of Missions

A 3D simulation environment named $NEMO_{CAT}$ (Antich and Ortiz, 2004) was used to measure the path length of the two first navigational approaches as well as T^2. This simulator incorporates the dynamic model of a real underwater vehicle called GARBI, designed and built by the Computer Vision and Robotics research group of the University of Girona (Spain), making thus the simulations more realistic.

Initially, three environments were defined for testing purposes (see fig. 11). In the first one, walls/rocks of different length impede the progress of the vehicle towards its goal. The second environment, on the other hand, corresponds to a very deep box-shaped canyon. Finally, the third one appeared in (Ranganathan and Koenig, 2003), where a control system with deliberative capabilities was employed to solve it. *Avoiding the Past* and *LM* strategies were not able to successfully carry out any of the previously described missions, which shows their poor effectiveness to escape from large trapping areas. In both cases, the simulation was stopped after a travel time twice the longest of T^2.

In order to continue with the comparative study, a robot programming environment based on the AuRA (Arkin and Balch, 1997) architecture called *MissionLab* (Mackenzie *et al.*, 1997) was also used. The latest release of this software (version 6.0) integrates the μNAV algorithm implemented by one of its authors (Sgorbissa, 2000). Different tests with increasing complexity were performed in *MissionLab*, simulating a

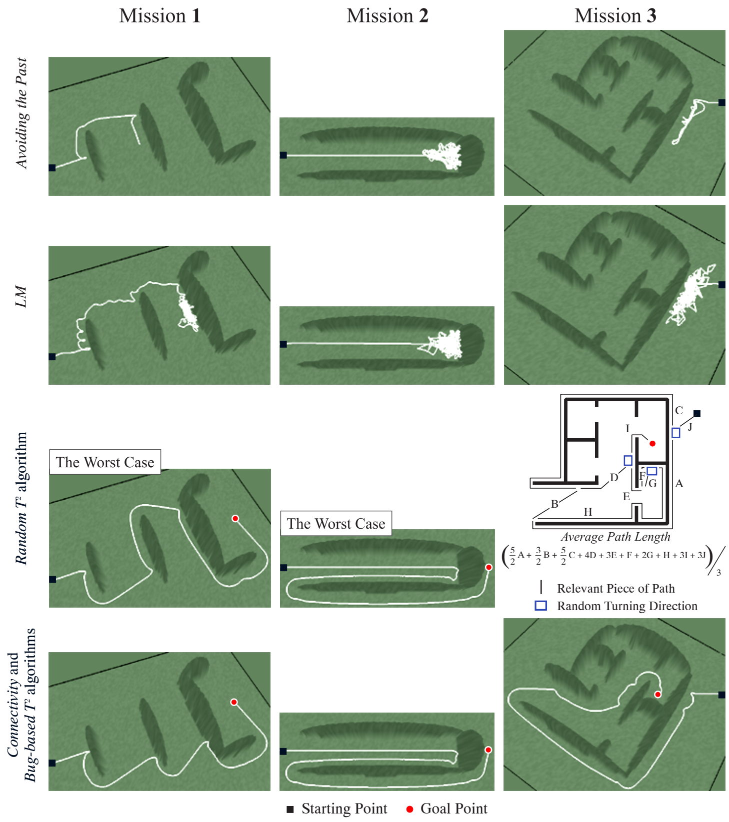


Fig. 11. From top to bottom, paths generated by the *Avoiding the Past*, *LM*, and T^2 approaches in $NEMO_{CAT}$. Note that due to the non-deterministic behavior of the Random T^2 algorithm, different results can be obtained in different runs of the simulator. Only one, the worst, is shown in the figure whenever possible. A stochastic analysis about the average length of the robot's path is presented when such worst case cannot be computed. For more details, the reader is referred to (Antich, 2006).

holonomic robot equipped with several range finders, and wheel encoders to compute its position by means of dead-reckoning. As can be verified in (Scalzo *et al.*, 2003), such experiments are a representative sample of the whole power of the μNAV strategy for a typical behavior hierarchy. Each environment was then accurately reproduced in $NEMO_{CAT}$ and successfully solved by T^2. The results from *MissionLab* are shown in fig. 12(a) while fig. 12(b) provides the ones from $NEMO_{CAT}$. Besides, tables 1 and 2 compare the performance of all these strategies from the viewpoint of the resultant path length. As can be observed, the T^2 algo-

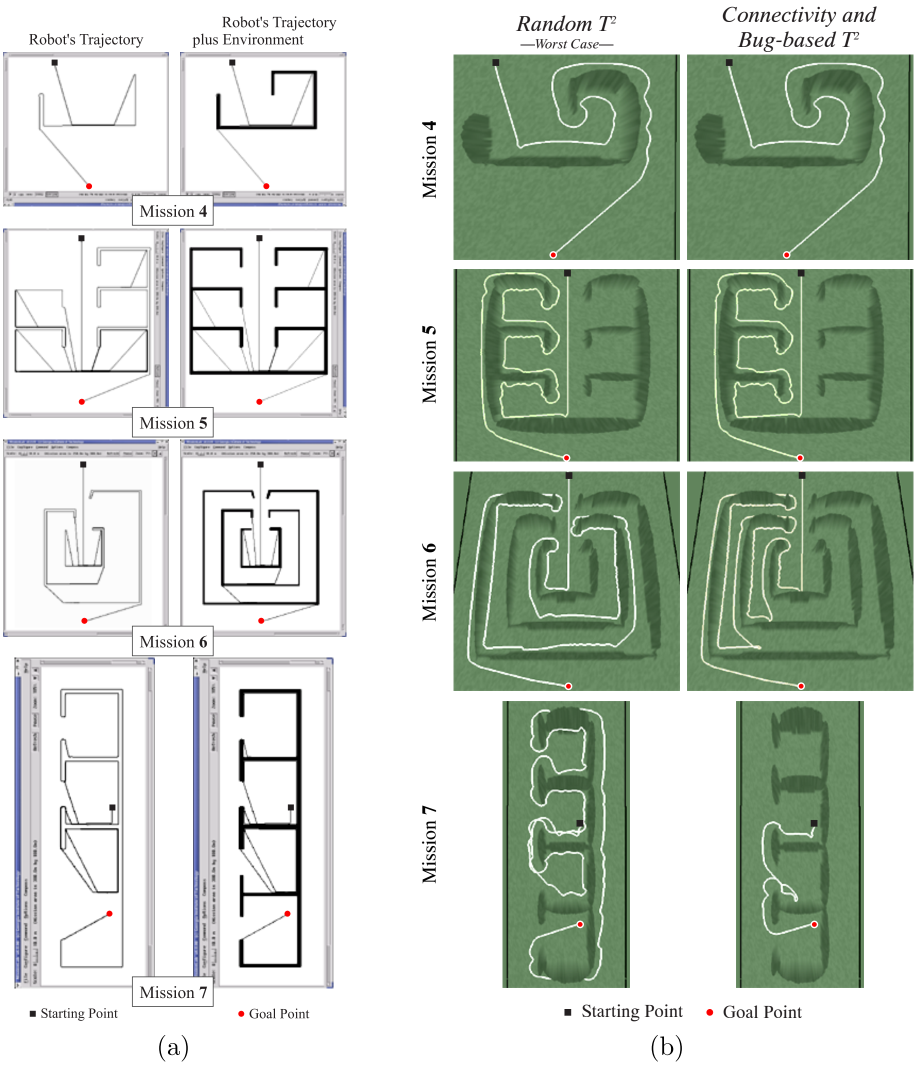


Fig. 12. Simulation results for the μNAV(a) and T^2(b) strategies in four different scenarios where concavities appear.

rithms produced, on average, trajectories between the starting and goal points 2.4 times shorter than μNAV at worst. The difference derives from the fact that μNAV allows the robot, in general, to head for the goal as soon as it is faced without any immediate obstacle on its way, while any T^2-based strategy limits the applicability of such rule to, at least, situations where a concavity is not detected.

Finally, the theoretical/ideal path of the *Bug2* algorithm for the missions previously outlined was drawn by hand according to the steps described in (Lumelsky and Stepanov, 1987) (see fig. 13). Tables 3 and 4 provide the comparative data with regard to our proposal. As can be seen, the *Bug2* algorithm generated, on average, trajectories between 1.14 and 1.48 times longer than T^2. On this occasion, the strict condition associated with the end of the contour following process is the key cause of the lower performance of such algorithm. Only when the main line is cut by the robot's path, the *GoTo* behavior becomes active to ensure convergence.

6 Conclusions

In this chapter, based on both artificial potential fields and two new principles named *Traversability* and *Tenacity* (T^2), a novel family of geometric algorithms for sensor-based motion planning has been finally put forward trying to ensure, whenever possible, the global achievement of the target point for any mission. The members of this family are specifically called *Random* T^2, *Connectivity* T^2 and *Bug-based* T^2. These strategies have also been compared against other well-known algorithms —*Avoiding the Past*, *Learning Momentum*, *Micronavigation* and *Bug2*— sharing the same goal. The length of the resulting paths was used as the figure of merit. Our proposals generated, on average, trajectories between 1.14 and 3.20 times shorter for a representative set of missions.

Finally, it is important to note that any T^2-based strategy can be applied to both ground and underwater vehicles, although, in this last case, a generalization to three dimensions, which is being developed at the moment, is expected to yield still better performance.

Acknowledgments

This study has been partially supported by project CICYTDPI2005-09001-C03-02 and FEDER funds. Its authors wish to thank the Georgia Tech Research Corporation for the free distribution of its software *MissionLab*.

Table 1. Comparison of the Path Lengths of T^2 and the μNAV Strategy

Mission	Algorithm Type		
	T^2		
	Random (average)	*Connectivity* and *Bug-based*	μNAV
4	286.83	368.55	384.20
5	883.47	883.47	4180.41
6	1586.71	1484.91	2489.40
7	663.82	246.85	1316.40
Total (m)	3420.83	2983.78	8370.41

Table 2. Relative Performance of T^2 with regard to the μNAV Strategy

Mission	$\frac{\mu\text{NAV}}{\text{Random } T^2}$	$\frac{\mu\text{NAV}}{\text{Connectivity } T^2}, \frac{\mu\text{NAV}}{\text{Bug-based } T^2}$
4	1.34	1.04
5	4.73	4.73
6	1.57	1.68
7	1.98	5.33
Average	2.40	3.20

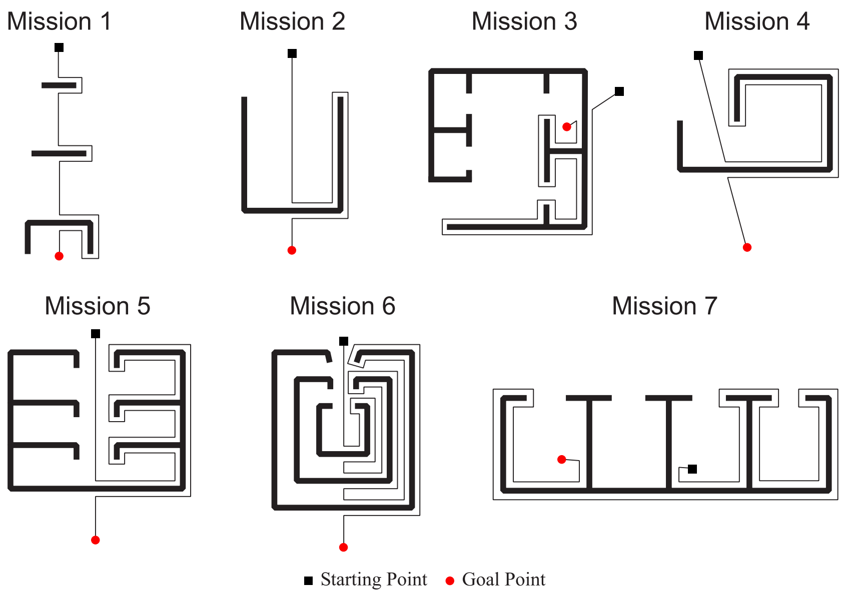


Fig. 13. Expected results for the *Bug2* algorithm from mission 1 to 7.

Table 3. Comparison of the Path Lengths of T^2 and the *Bug2* Strategy

Mission	Algorithm Type		
	T^2		
	Random (average)	*Connectivity* and *Bug-based*	*Bug2*
1	313.62	281.54	388.00
2	420.65	420.65	426.00
3	623.00	405.17	578.00
4	286.83	368.55	421.00
5	883.47	883.47	934.00
6	1586.71	1484.91	1672.00
7	663.82	246.85	797.00
Total (m)	4778.10	4091.14	5216.00

Table 4. Relative Performance of T^2 with regard to the *Bug2* Strategy

Mission	$\frac{\text{Bug2}}{\text{Random } T^2}$	$\frac{\text{Bug2}}{\text{Connectivity } T^2}, \frac{\text{Bug2}}{\text{Bug-based } T^2}$
1	1.24	1.38
2	1.01	1.01
3	0.93	1.43
4	1.47	1.14
5	1.06	1.06
6	1.05	1.13
7	1.20	3.23
Average	1.14	1.48

References

Antich, J. 2006. Reactive Robotics: A Paradigm Not Limited to Simple Tasks. Master's thesis, University of the Balearic Islands.

Antich, J. and Ortiz, A. 2004. An underwater simulation environment for testing autonomous robot control architectures. In Proceedings of the Conference on Control Applications in Marine Systems, pages 509–514.

Arkin, R. and Balch, T. 1997. AuRA: Principles and practice in review. Journal of Experimental and Theoretical Artificial Intelligence, **9**(2-3): 175–188.

Balch, T. and Arkin, R. 1993. Avoiding the past: a simple but effective strategy for reactive navigation. In Proceedings of the International Conference on Robotics and Automation, pages 678–685.

Khatib, O. 1986. Real-time obstacle avoidance for manipulators and mobile robots. International Journal of Robotics Research, **5**(1): 90–98.

Koren, Y. and Borenstein, J. 1991. Potential field methods and their inherent limitations for mobile robot navigation. In Proceedings of the International Conference on Robotics and Automation, pages 1398–1404.

Lee, J. and Arkin, R. 2001. Learning momentum: integration and experimentation. In Proceedings of the International Conference on Robotics and Automation, pages 1975–1980.

Lumelsky, V. and Stepanov, A. 1987. Path planning strategies for a point mobile automaton moving amidst unknown obstacles of arbitrary shape. Algorithmica, **2**: 403–430.

Mackenzie, D., Arkin, R., and Cameron, J. 1997. Multiagent mission specification and execution. Autonomous Robots, **4**(1): 29–52.

Ranganathan, A. and Koenig, S. 2003. A reactive robot architecture with planning on demand. In Proceedings of the International Conference on Intelligent Robots and Systems, pages 1462–1468.

Scalzo, A., Sgorbissa, A., and Zaccaria, R. 2003. μNAV: a minimalist approach to navigation. In Proceedings of the International Conference on Robotics and Automation, pages 2018–2023.

Sgorbissa, A. 2000. Toward a Multi-Ethnic Community of Humans, Mobile Robots, and Intelligent Devices. Ph.D. thesis, Università di Genova.

CHAPTER 8

Underwater climbing robot for ships' hull cleaning

R. FERNÁNDEZ, M. ARMADA, T. AKINFIEV, and P. GONZÁLEZ DE SANTOS

Instituto de Automática Industrial - CSIC roemi@iai.csic.es, armada@iai.csic.es, teodor@iai.csic.es, pgds@iai.csic.es

In this chapter, AURORA project is introduced and a solution is proposed to improve the performance of an underwater climbing robot that has been designed for cleaning and surveying tasks on ship hulls. The solution implies a non-classically actuated wheeled mobile robot that has as many traction motors as wheels controlled independently. It also implies the utilization of a mixed velocity-force control algorithm. In this way, an increase in traction force is achieved and the sliding of the wheels during the movement process is prevented. The effectiveness of the new algorithm is demonstrated in laboratory experiments using a prototype.

1 Introduction

It is well known that all kind of underwater ship's hull become overgrown with sea adherence (weed, barnacles) very fast. This means raise of fuel consumption, and freeing atmosphere an extra amount of CO2 (incrementing greenhouse effect) and of sulphur dioxide (acid rain), apart from deterioration of ability of ship's control. This situation becomes important even after six months of ship activity. For recovery of ship's required operational performance, it is necessary for Ship-Repairing and Conversion Industries to dry dock a ship and proceed to cleaning. This procedure is very time consuming and of high cost, but it is the only available solution nowadays for SRYs. On the other hand, this cleaning activity is the first to be done when a ship needs maintenance and/or some repairing, being the last the main activity of SRYs. So hull treatment is required and, at present time, is done manually in dry-dock using different adapted methods like grit blasting or water jet, and it has to be noticed that, in itself, it is a very contaminant operation (dust contains always painting particles), it is harmful for human operators health and it is a very uncomfortable job. To provide a solution to these problems an EC funded project (G3RD-CT-000-00246) was organised: AURORA (Auxiliary Climbing Robot for Underwater Ship Hull Cleaning of Sea Adherence and Surveying). The AURORA scenario consists in the underwater hull that after some time of ship operation is plenty of marine incrustations, where a new kind of underwater climbing robot equipped with special tools should perform cleaning and surveying tasks. That scenario (see Fig. 1) presents large dimensions and exhibits some areas of very difficult reach-ability that poses some additional technical difficulties. The proposed robot has three wheels for balance reasons, and it is

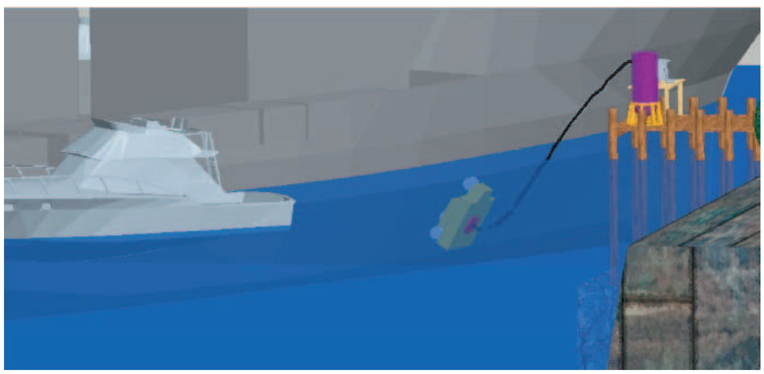


Fig. 1. Aurora scenario.

endowed with magnets to attach to ferromagnetic surfaces. Although in the last decade much effort has been oriented to solve the motion problem of a wheeled mobile robot under nonholonomic constraints, special considerations have to be taken into account here in the development of the control system because of both the operating environment and the requirements of the cleaning process. Thus, to achieve a proper quality of cleaning, the robot has to move with a suitable velocity, following some kind of strategy to cover the foul zone. This leads to the importance of controlling the direction of the robot and its velocity. On the other hand, the processed surface is characterized with a very low magnitude of static friction coefficient, causing difficulty to obtain an appropriated magnitude of traction force. To solve this problem, it has been necessary to increase the traction force but preventing the wheels from sliding during the movement process. Traction force has been easily increased by having as many power motors for traction as wheels controlled independently. However, having such configuration, quite far from the classical ones, a new control algorithm is needed. This velocity-force control algorithm has to sustain the required velocity for the technological process, and, at the same time, distributes the traction forces between the wheels (Akinfiev *et al.*, 2001), (Fernández *et al.*, 2002). Next sections describe the main features of the underwater robot with special focus on its control system.

2 Kinematic Model

This section describes the kinematic equations of the three-wheeled mobile robot depicted in Fig. 2. The common configurations for a three-wheeled mobile robot includes two motors: one power motor and one steering motor on the front wheel (Everett, 1995), (Bestaoui, 2000), or two power motors on the rear wheels, with the front wheel having passive steering (Aguiar *et al.*, 2000), (Fukao *et al.*, 2000), (Ge *et al.*, 2001). Basically, both configurations share the same control and structural properties. Such designs demonstrate a good performance when moving upon a surface with an adequate coefficient of static friction and a smaller coefficient of rolling friction. However, in the case considered in this thesis, the wheeled mobile robot has to move along a surface that is characterized with a very low magnitude of static friction coefficient. Since the static friction force is equal to the traction force, the low magnitude of the static friction coefficient causes difficulty to obtain an appropriated magnitude of traction force for the wheeled mobile robot. In order to solve this problem, the proposed configuration for the mobile robot includes as many power motors for traction controlled individually as number of wheels. Then, the vehicle has three identical wheels which are controlled by three DC motors. The front wheel is also steerable by an additional DC motor. It is assumed that the masses and inertias of the wheels are negligible. It is further assumed that the contact between the wheels and the ground is pure rolling and non-slipping. This last assumption imposes the nonholonomic constraint.

The position of the wheeled mobile robot along a flat or quasi-flat surface in an inertial Cartesian frame $\{R_w\} := \{o, x, y\}$ is completely specified by the (x_C, y_C) coordinates of the reference point R_p and θ, the orientation of the vehicle with respect to the x-axis. R_p is assumed to be

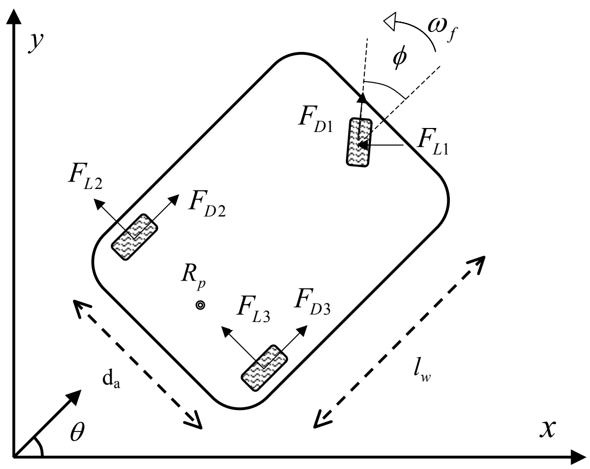


Fig. 2. Wheeled mobile robot's configuration.

located in the intersection of the virtual axis that links the rear wheels and the axis of symmetry. Let ϕ denote the steering angle between the front wheel an the symmetry axis, d_a the length of the virtual axis that links the rear wheels, l_w the distance between the front wheel and R_p, (x_1, y_1) the position of the front wheel, (x_2, y_2) and (x_3, y_3) the position of the rear wheels. Then, the position of each wheel in the inertial reference frame may be stated as:

$$\begin{aligned} x_1 &= x_C + l_w \cos(\theta), \\ y_1 &= y_C + l_w \sin(\theta), \end{aligned} \tag{1}$$

$$\begin{aligned} x_2 &= x_C - \frac{d_a}{2} \sin(\theta), \\ y_2 &= y_C + \frac{d_a}{2} \cos(\theta), \end{aligned} \tag{2}$$

$$\begin{aligned} x_3 &= x_C + \frac{d_a}{2} \sin(\theta), \\ y_3 &= y_C - \frac{d_a}{2} \cos(\theta). \end{aligned} \tag{3}$$

Their derivatives are given by:

$$\begin{aligned} \dot{x}_1 &= \dot{x}_C - l_w \dot{\theta} \sin(\theta), \\ \dot{y}_1 &= \dot{y}_C + l_w \dot{\theta} \cos(\theta), \end{aligned} \tag{4}$$

$$\begin{aligned} \dot{x}_2 &= \dot{x}_C - \frac{d_a}{2} \dot{\theta} \cos(\theta), \\ \dot{y}_2 &= \dot{y}_C - \frac{d_a}{2} \dot{\theta} \sin(\theta), \end{aligned} \tag{5}$$

$$\begin{aligned} \dot{x}_3 &= \dot{x}_C + \frac{d_a}{2} \cos(\theta), \\ \dot{y}_3 &= \dot{y}_C + \frac{d_a}{2} \sin(\theta). \end{aligned} \tag{6}$$

Then, the nonholonomic constraints are written for each wheel, resulting in:

$$\dot{x}_1 \sin(\theta + \phi) - \dot{y}_1 \cos(\theta + \phi) = 0, \tag{7}$$

$$\dot{x}_2 \sin(\theta) - \dot{y}_2 \cos(\theta) = 0, \tag{8}$$

$$\dot{x}_3 \sin(\theta) - \dot{y}_3 \cos(\theta) = 0. \tag{9}$$

Substituting $(\dot{x}_1, \dot{y}_1)$, $(\dot{x}_2, \dot{y}_2)$ and $(\dot{x}_3, \dot{y}_3)$ into equations (7), (8) and (9) gives:

$$\dot{x}_C \sin(\theta + \phi) - \dot{y}_C \cos(\theta + \phi) - l_w \dot{\theta} \cos(\phi) = 0, \tag{10}$$

$$\dot{x}_C \sin(\theta) - \dot{y}_C \cos(\theta) = 0. \tag{11}$$

Now, let v_c denote the linear velocity of R_p with respect to $\{R_w\}$. Then the linear velocities of the center of each wheel with respect to $\{R_w\}$ are given by (see Fig. 3):

$$v_1 = \frac{v_c}{\cos\phi}, \tag{12}$$

$$v_2 = \left(1 - \frac{d_a}{2l_w} \tan\phi\right) v_c, \tag{13}$$

$$v_3 = \left(1 + \frac{d_a}{2l_w} \tan\phi\right) v_c. \tag{14}$$

$\dot{x}_C$ and $\dot{y}_C$ can also be defined as a function of v_c as follows:

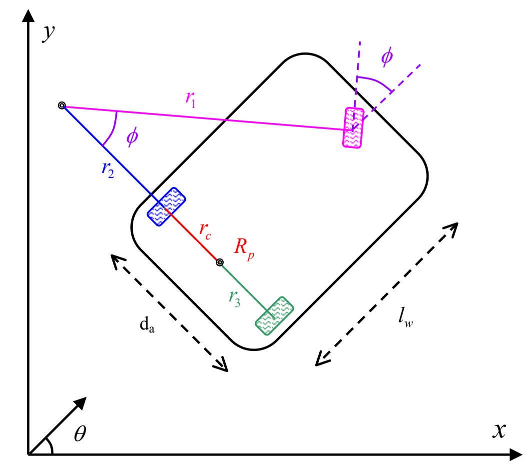


Fig. 3. Wheeled mobile robot.

$$\dot{x}_C = v_c \cos(\theta), \tag{15}$$

$$\dot{y}_C = v_c \sin(\theta). \tag{16}$$

Substituting (15), (16), into equations (10) and (11) results in:

$$\dot{\theta} = \frac{\tan\phi}{l_w} v_c. \tag{17}$$

The derivative of (17) is given by:

$$\ddot{\theta} = \frac{\tan(\phi)}{l_w}\dot{v}_c + \frac{v_c}{l_w \cos^2(\phi)}\dot{\phi}. \tag{18}$$

If we suppose that the kinematic model of the wheeled mobile robot is composed of equations (15-17), we had the steering angle ϕ as an input and assumed to have direct control over its value. This corresponds to being able to move the front wheel instantaneously. Nevertheless, this assumption is an unrealistic one. Whenever the value of ϕ changes discontinuously, the path traced out in the plane by (x_c, y_c), will have a discontinuity in its curvature. To put this right, we might add the steering angle as an extra state variable and consider its derivative to be one of the input signals. Introducing this integrator-chain, ϕ is only allowed to change its values in a continuous manner. Thus, a four-dimensional state space has to be considered, in which each state is represented as

$x = (x_c, y_c, \theta, \phi)$. The kinematic model of the three-wheeled mobile robot is then given by the following equations:

$$\begin{aligned} \dot{x}_C &= v_c \cos(\theta), \\ \dot{y}_C &= v_c \sin(\theta), \\ \dot{\theta} &= \frac{\tan(\phi)}{l_w} v_c, \\ \dot{\phi} &= \omega_f, \end{aligned} \tag{19}$$

where ω_f is the angular steering velocity of the front wheel.

3 Dynamic Model

This section deals with system dynamics. It is assumed that there is no friction force between the wheels and the vehicle, and that the axis of the rear wheels are locked to be in the same orientation as the mobile robot. The resultant dynamic equations are:

$$\begin{aligned} \sum F_x = {} & F_{D1} \cos(\phi + \theta) + (F_{D2} + F_{D3}) \cos(\theta) - F_{L1} \sin(\phi + \theta) - \\ & - (F_{L2} + F_{L3}) \sin(\theta) - F_{FRW} \cos(\theta), \end{aligned} \tag{20}$$

$$\begin{aligned} \sum F_y = {} & F_{D1} \sin(\phi + \theta) + (F_{D2} + F_{D3}) \sin(\theta) + F_{L1} \cos(\phi + \theta) + \\ & + (F_{L2} + F_{L3}) \cos(\theta) - F_{FRW} \sin(\theta), \end{aligned} \tag{21}$$

$$\sum M_z = \frac{d_a}{2} (F_{D3} - F_{D2}) + l F_{D1} \sin(\phi) + l_w F_{L1} \cos(\theta) - M_{FR}, \tag{22}$$

where F_{D1}, F_{D2} and F_{D3} are the traction forces applied at each wheels, F_{L1}, F_{L2} and F_{L3} are the resultant lateral forces on the front and rear tires respectively, F_{Fr} is the equivalent force of friction, and M_{FR} is the equivalent moment of friction around R_p. Since the velocity of the wheeled mobile robot is quite low, the lateral forces, F_{L1}, F_{L2} and F_{L3} are neglected in the sequel because of their small influence. It is important to remark that other forces can appear in the dynamic model (i.e., hydrodynamic friction, if the robot moves underwater; forces due to gravity, if the wheeled mobile robot moves along a vertical wall or an inclined plane; an additional resistance force, if the robot performs cleaning tasks). In order to simplify the dynamic model without loss of generality, we assume that the influence of all these forces are contained in the friction terms F_{FRW} and M_{FR}. Now, taking into account that:

$$\sum M_z = J_w \ddot{\theta}, \tag{23}$$

where J_w is the equivalent inertia of the vehicle around R_p, and placing (18) and (23) into equation (22) gives:

$$\begin{aligned}\dot{v}_c = {} & \frac{l_w}{J_w} \cot(\phi) \left[l F_{D1} \sin(\phi) + \frac{d_a}{2} (F_{D3} - F_{D2}) - M_{FR} \right] \\ & - \dot{\phi} \cot(\phi) \sec^2(\phi) \, v_c.\end{aligned} \tag{24}$$

The next step in deriving the dynamic model of the system is to find equations for the traction forces. Since it is known that the traction forces are transmitted to the vehicle through controlled DC motors, they are modelled as follows:

$$F_{D1} = \frac{K_G}{r_w} \tau_{ap1}, \tag{25}$$

$$F_{D2} = \frac{K_G}{r_w} \lambda_{w2} \tau_{ap1}, \tag{26}$$

$$F_{D3} = \frac{K_G}{r_w} \lambda_{w3} \tau_{ap1}, \tag{27}$$

where K_G is the constant transmission ratio, r_w is the wheel radius, τ_{ap1} is the traction torque of the front wheel, and λ_{w2} and λ_{w3} are traction torque distribution coefficients. These coefficients, λ_{w2} and λ_{w3}, play a crucial role in the development of the control system, and their inclusion will be explained in detail in the subsections 4.2 and 4.3. The traction torque of the front wheel τ_{ap1} is given by:

$$\tau_{ap1} = k_m I_1 - J_{M1} \ddot{\varphi}_1, \tag{28}$$

where k_m is the torque constant, I_1 is the armature current, J_{M1} is the rotor inertia and $\ddot{\varphi}_1$ is the angular acceleration of the rotor. The armature current, I_1, is given by:

$$I_1 = \frac{1}{R_M} u_1 - \frac{k_E}{R_M} \dot{\varphi}_1, \tag{29}$$

where u_1 is the input voltage for the traction motor of the front wheel, R_M is the motor resistance, and k_E is the back - EMF constant.

The relationship between the linear velocity of the center of the front wheel and the angular velocity of its motor may be stated as follows:

$$\dot{x}_1 = \frac{r_w}{K_G} \dot{\varphi}_1 \cos(\phi + \theta), \tag{30}$$

$$\dot{y}_1 = \frac{r_w}{K_G} \dot{\varphi}_1 \sin(\phi + \theta). \tag{31}$$

Combining equations (4), (15), (16), (30) and (31), the next equalities are obtained:

$$\begin{aligned} v_c \cos(\theta) - l_w \dot{\theta} \sin(\theta) &= \frac{r_w}{K_G} \dot{\varphi}_1 \cos(\phi + \theta), \\ v_c \sin(\theta) + l_w \dot{\theta} \cos(\theta) &= \frac{r_w}{K_G} \dot{\varphi}_1 \sin(\phi + \theta). \end{aligned} \tag{32}$$

Then, from the previous equalities, the relationship between the linear velocity of the reference point, v_c, and the angular velocity of the rotor of the front wheel, $\dot{\varphi}_1$, can be found:

$$\dot{\varphi}_1 = \frac{K_G}{r_w \cos(\phi)} v_c. \tag{33}$$

The derivative of (33) is given by:

$$\ddot{\varphi}_1 = \frac{K_G}{r_w \cos(\phi)} \left[\dot{\phi} \tan(\phi) v_c + \dot{v}_c \right]. \tag{34}$$

Substituting (25-27), (28) and (33-34) into (24) results in:

$$\begin{aligned} \dot{v}_c = {} & \frac{\gamma_{wa}}{\gamma_{wb}} \left[\frac{K_G k_m}{r_w R_M} u_1 - \frac{K_G^2 v_c}{r_w^2 \cos(\phi)} \left(\frac{k_m k_E}{R_M} + J_{M1} \omega_f \tan(\phi) \right) \right] \\ & - \frac{1}{\gamma_{wb}} \left[M_{FR} + \frac{J_w}{l_w} \omega_f \sec^2(\phi) v_c \right], \end{aligned} \tag{35}$$

where:

$$\gamma_{wa} = \frac{d_a}{2} (\lambda_{w3} - \lambda_{w2}) + l_w \sin(\phi), \tag{36}$$

$$\gamma_{wb} = \frac{J_w}{l_w} \tan(\phi) + \frac{K_G^2 J_{M1}}{r_w^2 \cos(\phi)} \gamma_{wa}. \tag{37}$$

The final step is to find the equation for the steering servo. Since the steering force is also transmitted to the vehicle trough a controlled DC, it is modelled as follows:

$$J_{M4}\ddot{\varphi}_4 = \frac{k_m}{R_M} (u_4 - k_E \dot{\varphi}_4) - \tau_{FR4}, \tag{38}$$

where J_{M4} is the rotor inertia, $\ddot{\varphi}_4$ is the angular acceleration of the rotor, $\dot{\varphi}_4$ is the angular velocity of the rotor, u_4 is the input voltage for the steering motor of the front wheel and τ_{FR4} is the torque of friction. In this case, the relationship between the angular velocity of the front wheel and the angular velocity of its motor is directly stated as follows:

$$\dot{\phi} = \frac{\dot{\varphi}_4}{K_G}. \tag{39}$$

Taking into account that $\dot{\phi} = \omega_f$ and using equations (38-39), the angular acceleration of the front wheel, $\dot{\omega}_f$, can be written as:

$$\dot{\omega}_f = \frac{1}{J_{M4} K_G} \left(\frac{k_m}{R_M} u_4 - \frac{K_G k_m k_E \omega_f}{R_M} + \tau_{FR4} \right). \tag{40}$$

Then, the dynamics of the wheeled mobile robot is modelled by the equations (35) and (40).

4 Control Algorithms

Now that the whole mathematical model has been identified, the control system for the wheeled mobile robot can be designed. This task is quite complicated, not only because of the nonholonomic constraints that appear in the kinematics, but also due the special requirements of the desired application. The objective is to control the position and the velocity of the wheeled mobile robot but, at the same time, to distribute the traction force between the wheels in such a way that their sliding is avoided. The problem is even paradoxical: the prevention of sliding on the wheels compels the system to be nonholonomic, which augments the complexity of the control system design. To overcome these difficulties, the control design is divided in three related parts. Firstly, a state-feedback tracking controller is derived to solve the motion problem under nonholonomic constraints using the kinematic model of the mobile robot. Secondly, an extension of the kinematic control law is made to incorporate the dynamics of the wheeled mobile robot in such a way that the asymptotic stability that was originally obtained by the kinematic controller is preserved for the overall closed loop system. Thirdly, a traction force distribution algorithm that calculates the proper reference signals for each rear wheel is included and the feedback tracking control laws are finally completed.

4.1 Kinematic Control

In the methodology proposed above for the control system design, the first step is to obtain a state-feedback tracking controller for the kinematic model (19) of the wheeled mobile robot. Since this issue has been extensively discussed by a large number of authors, we will follow the method proposed in (Lefer and Nijmeijer, 1999) and their results. Then, the forward velocity of the reference point, v_c, and the angular velocity of the front wheel, ω_f, from the kinematic model of the robot (19), are considered as control inputs. Given a feasible reference trajectory, i.e. a trajectory $\left([x_r, y_r, \theta_r, \phi_r]^T, [v_r, \omega_r]^T \right)$ satisfying:

$$\begin{aligned} \dot{x}_r &= v_r \cos(\theta_r), \\ \dot{y}_r &= v_r \sin(\theta_r), \\ \dot{\theta}_r &= v_r \frac{\tan(\phi_r)}{l_w}, \\ \dot{\phi}_r &= \omega_r, \end{aligned} \tag{41}$$

it is necessary to find appropriate control laws, v_c and ω_f, of the form:

$$v_c = v_c(x_C, y_C, \theta, \phi, t), \qquad \omega_f = \omega_f(x_C, y_C, \theta, \phi, t), \tag{42}$$

such that for the resulting closed-loop system (19, 42):

$$\begin{aligned} \lim_{t\to\infty} (&|x_C(t) - x_r(t)| + |y_C(t) - y_r(t)| + \\ &+ |\theta(t) - \theta_r(t)| + |\phi(t) - \phi_r(t)|) \quad = 0. \end{aligned} \tag{43}$$

The control laws (42) are not only a function of x_C, y_C, θ and ϕ, but also of $v_r(t)$, $\omega_r(t)$, $x_r(t)$, $y_r(t)$, $\theta_r(t)$, $\phi_r(t)$, and possibly their derivatives with respect to time. This explains the time-dependency in (42).

Notice that the tracking control problem studied here is not the same as an output tracking problem of the flat output $\left([x_r(t), y_r(t)]^T\right)$. First of all, by specifying $x_r(t)$ and $y_r(t)$ the reference trajectory can not be uniquely specified (i.e. $v_r(t)$ can be either positive or negative). But more important is the fact that tracking of $x_r(t)$ and $y_r(t)$ does not guarantee tracking of the corresponding $\theta_r(t)$ and $\phi_r(t)$ (Lefer and Nijmeijer, 1999).

In order to be able to solve the tracking control problem, the following assumptions on the reference trajectory are needed:

- The reference system needs to have a unique solution. Thus, $\phi_r \in\ ]-M, M[$ with $M < \frac{\pi}{2}$.
- We assume that the forward reference velocity is bounded, i.e. there exists constants $v_r^{\min}$ and $v_r^{\max}$ such that $0 < v_r^{\min} \leq v_r(t) \leq v_r^{\max}$.
- Furthermore, we assume that the forward and angular acceleration, i.e. $\dot{v}_r$ and $\ddot{\theta}_r$, are bounded.

Remark 1. Throughout this section the expressions $\frac{x\cos(x)-\sin(x)}{x^2}$, $\frac{x-\sin(x)}{x^2}$, $\frac{\cos(x)-1}{x}$, $\frac{x\sin(x)-2(1-\cos(x))}{x^3}$, $\frac{(1-\cos(x))x-2(x-\sin(x))}{x^3}$, $\frac{1-x\sin(x)-\cos(x)}{x^2}$, $\frac{\cos(x)-1}{x^2}$ and $\frac{\sin(x)}{x}$, are used. These functions are discontinuous in $x=0$, but if their values are defined for $x=0$ as respectively 0, 0, 0, 0, $\frac{1}{6}$, $-\frac{1}{2}$, $-\frac{1}{2}$ and 1 it is easy to verify that all the functions are continuous and bounded.

To overcome the problem that the errors $x_C - x_r$ and $y_C - y_r$ depend on how we choose the inertial reference frame, we define errors in a body reference frame, i.e. in a coordinate-frame attached to the wheeled mobile robot (Kanayama *et al.*, 1990):

$$\begin{bmatrix} x_e \\ y_e \\ \theta_e \end{bmatrix} = \begin{bmatrix} \cos(\theta) & \sin(\theta) & 0 \\ -\sin(\theta) & \cos(\theta) & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x_r - x_C \\ y_r - y_C \\ \theta_r - \theta \end{bmatrix}. \tag{44}$$

In order to be able to control the orientation, θ, of the mobile robot by means of the input ω_f, Lefeber (Lefer and Nijmeijer, 1999) proposed to have $v_c(t) \neq 0$ for all $t \geq 0$. Since $v_r(t) \geq v_r^{\min} > 0$, if $\sigma(\cdot)$ is a function that fulfills:

$$\sigma(x) > -v_r^{\min}, \qquad \forall x \in \mathbb{R}, \tag{45}$$

the control law:

$$v_c = v_r + \sigma(x_e) \equiv \Phi_{w1}(t, x_e), \tag{46}$$

automatically guarantees $v_c(t) > 0$ for all $t \geq 0$. Furthermore, we assume that $\sigma(x)$ is continuously differentiable and satisfies:

$$x\sigma(x) > 0, \forall x \neq 0. \tag{47}$$

In our case, we choose:

$$\sigma(x_e) = v_r^{\min} \tanh(x_e). \tag{48}$$

With the control law (46), the error system (44) and ϕ become:

$$\begin{aligned} \dot{x}_e &= y_e \frac{v_r + \sigma(x_e)}{l_w} \tan(\phi) + v_r \cos(\theta_e) - v_r - \sigma(x_e), \\ \dot{y}_e &= -x_e \frac{v_r + \sigma(x_e)}{l_w} \tan(\phi) + v_r \sin(\theta_e), \\ \dot{\theta}_e &= \frac{v_r}{l_w} \tan(\phi_r) - \frac{v_r + \sigma(x_e)}{l_w} \tan(\phi), \\ \dot{\phi} &= \omega_f. \end{aligned} \tag{49}$$

Differentiating the function:

$$V_1 = \frac{1}{2} x_e^2 + \frac{1}{2} y_e^2 \tag{50}$$

along the solution of (49) yields:

$$\dot{V}_1 = -x_e \sigma(x_e) + v_r \left(\frac{\cos(\theta_e) - 1}{\theta_e} x_e + \frac{\sin(\theta_e)}{\theta_e} y_e \right) \theta_e. \tag{51}$$

Now, if ϕ is considered as a virtual control, an intermediate control law for ϕ could be designed:

$$\dot{\theta}_e = -k_{w1}\theta_e - k_{w2} v_r \left(\frac{\cos(\theta_e) - 1}{\theta_e} x_e + \frac{\sin(\theta_e)}{\theta_e} y_e \right) \qquad c_{w1}, c_{w2} > 0. \tag{52}$$

Using the Lyapunov function candidate:

$$V_2 = \frac{1}{2} x_e^2 + \frac{1}{2} y_e^2 + \frac{1}{2k_{w2}} \theta_e^2, \tag{53}$$

we can then assert that x_e, y_e and θ_e converge to zero, provided that assumption 4.1 is satisfied.

Now, the error variable is defined as follows:

$$\bar{z} = \frac{v_r}{l_w}\tan(\phi_r) - \frac{v_c}{l_w}\tan(\phi) + k_{w1}\theta_e$$
$$+k_{w2}v_r\left(\frac{\cos(\theta_e)-1}{\theta_e}x_e + \frac{\sin(\theta_e)}{\theta_e}y_e\right). \tag{54}$$

Nevertheless, for simplicity of analysis, a new error variable is defined:

$$z = v_r\tan(\phi_r) - v_c\tan(\phi) + c_{w1}\theta_e$$
$$+c_{w2}v_r\left(\frac{\cos(\theta_e)-1}{\theta_e}x_e + \frac{\sin(\theta_e)}{\theta_e}y_e\right), \tag{55}$$

where:

$$z = l_w\bar{z}, \qquad c_{w1} = l_w k_{w1}, \qquad c_{w2} = l_w k_{w2}. \tag{56}$$

With this definition, the error system (49) becomes:

$$\begin{aligned}
\dot{x}_e &= y_e\frac{v_c}{l_w}\tan(\phi) + v_r\left[\cos(\theta_e) - 1\right] - \sigma(x_e),\\
\dot{y}_e &= -x_e\frac{v_c}{l_w}\tan(\phi) + v_r\sin(\theta_e),\\
\dot{\theta}_e &= -\frac{c_{w1}}{l_w}\theta_e - \frac{c_{w2}}{l_w}v_r\beta_w + \frac{1}{l_w}z,\\
\dot{z} &= \frac{v_c}{\cos^2(\phi)}\left(\frac{1}{l_w}\iota_{w1}(t)\sin(\phi)\cos(\phi) - \omega_f\right) + \iota_{w2}(t),
\end{aligned} \tag{57}$$

where:

$$\iota_{w1} = y_e\tan(\phi) + c_{w1} - c_{w2}v_r\left(\alpha_{w1}x_e + \alpha_{w2}y_e\right), \tag{58}$$

$$\iota_{w2} = \dot{v}_r\tan(\phi_r) + \frac{v_r\omega_r}{\cos^2(\phi_r)} + \left(v_r\cos(\theta_e) - v_c\right)\tan(\phi) +$$
$$+c_{w2}\left(\dot{v}_r x_e - v_c v_r - v_r^2\right)\frac{\cos(\theta_e)-1}{\theta_e} + c_{w2}\dot{v}_r y_e\frac{\sin(\theta_e)}{\theta_e} + \tag{59}$$
$$\left(c_{w1} + c_{w2}v_r\alpha_{w3}\right)\frac{v_r}{l_w}\tan(\phi_r),$$

$$\alpha_{w1} = \frac{1-\cos(\theta_e)}{\theta_e^2}, \tag{60}$$

$$\alpha_{w2} = \frac{\theta_e - \sin(\theta_e)}{\theta_e^2}, \tag{61}$$

$$\alpha_{w3} = \left(\alpha_{w1} - \frac{\theta_e \sin(\theta_e)}{\theta_e^2}\right) x_e + \frac{\theta_e \cos(\theta_e) - \sin(\theta_e)}{\theta_e^2} y_e, \tag{62}$$

$$\beta_w = \frac{\cos(\theta_e) - 1}{\theta_e} x_e + \frac{\sin(\theta_e)}{\theta_e} y_e. \tag{63}$$

Then, a composite Lyapunov function V_3 is constructed for the system (57) by augmenting V_2 with a quadratic term in the error variable $\bar{z}$:

$$V_3 = \frac{1}{2} x_e^2 + \frac{1}{2} y_e^2 + \frac{l_w}{2c_{w2}} \theta_e^2 + \frac{1}{2c_{w1}c_{w2}c_{w3}} \bar{z}^2 \qquad \text{with} \qquad c_{w3} > 0. \tag{64}$$

Now, ω_f is chosen to make $\dot{V}_3$ negative definite. A simple way to achieve this is taking:

$$\omega_f = \frac{1}{l_w} \iota_{w1} \sin(\phi) \cos(\phi) + \frac{\cos^2(\phi)}{v_c} (\iota_{w2} + c_{w3} z) \equiv \Phi_{w2}(t, x_e, y_e, \theta_e). \tag{65}$$

With this control law, the CLF derivative is:

$$\dot{V}_3 = -x_e \sigma(x_e) - \frac{c_{w1}}{c_{w2}} \theta_e^2 + \frac{1}{c_{w2}} \theta_e z - \frac{1}{c_{w1}c_{w2}} z^2. \tag{66}$$

By completing the square, $\dot{V}_3$ can be rewritten as:

$$\begin{aligned} \dot{V}_3 = {} & -x_e \sigma(x_e) - \left[\frac{\sqrt{c_{w1}}}{\sqrt{2c_{w2}}} \theta_e - \frac{1}{\sqrt{2c_{w1}c_{w2}}} z\right]^2 - \frac{c_{w1}}{2c_{w2}} \theta_e^2 \\ & - \frac{1}{2c_{w1}c_{w2}} z^2. \end{aligned} \tag{67}$$

Then:

$$\dot{V}_3 \leq -x_e \sigma(x_e) - \frac{c_{w1}}{2c_{w2}} \theta_e^2 - \frac{1}{2c_{w1}c_{w2}} z^2 \leq 0. \tag{68}$$

From (67) it is clear that all trajectories of (57, 65) are globally uniformly bounded and all closed-loop solutions converge to zero (Lefer and Nijmeijer, 1999).

4.2 Dynamic Control

In the preceding subsection, the state-tracking problem for the kinematic model of the wheeled mobile robot was solved using velocity control inputs (v_c and ω_f). However, it is desirable to find feedback control laws for the actual control inputs u_1, u_2, u_3 and u_4. Then, in this subsection an extension of the resultant kinematic control laws is made to incorporate the dynamics of the wheeled mobile robot via backstepping. For that purpose, let us consider the tracking error dynamics of the wheeled mobile robot which is composed of (49) and two integrators (35, 40):

$$
\begin{aligned}
\dot{x}_e &= y_e \frac{v_r + \sigma(x_e)}{l_w} \tan(\phi) + v_r \cos(\theta_e) - v_r - \sigma(x_e), \\
\dot{y}_e &= -x_e \frac{v_r + \sigma(x_e)}{l_w} \tan(\phi) + v_r \sin(\theta_e), \\
\dot{\theta}_e &= \frac{v_r}{l_w} \tan(\phi_r) - \frac{v_r + \sigma(x_e)}{l_w} \tan(\phi), \\
\dot{\phi} &= \omega_f, \\
\dot{v}_c &= \frac{\gamma_{wa}}{\gamma_{wb}} \left[\frac{K_G k_m}{r_w R_M} u_1 - \frac{K_G^2 v_c}{r_w^2 \cos(\phi)} \left(\frac{k_m k_E}{R_M} + J_{M1} \omega_f \tan(\phi) \right) \right] \\
&\quad - \frac{1}{\gamma_{wb}} \left[M_{FR} + \frac{J_w}{l_w} \omega_f \sec^2(\phi)\, v_c \right], \\
\dot{\omega}_f &= \frac{1}{J_{M4} K_G} \left(\frac{k_m}{R_M} u_4 - \frac{K_G k_m k_E \omega_f}{R_M} + \tau_{FR4} \right).
\end{aligned}
\tag{69}
$$

The basic idea lies in deriving suitable feedback control laws for u_1 and u_4, from v_c and ω_f that control the kinematic system (19).

Remark 2. The traction force distribution coefficients, λ_{w2} and λ_{w3} (that are contained in the terms γ_{wa} and γ_{wb}), let us model the traction forces applied at the rear wheels, F_{D2} and F_{D3}, as some percent of the traction force applied at the front wheel, F_{D1}. Therefore, the feedback velocity control law for u_1 can be derived without neglecting the effects of u_2 and u_3. In the next subsection, feedback force control laws for u_2 and u_3 are derived using the proposed traction force distribution algorithm and the preservation of the asymptotic stability for the overall close loop system is demonstrated.

In order to design the true tracking controllers for u_1 and u_4 using the integrator backstepping, two new error variables, v_e and ω_e, are introduced:

$$
v_e = v_c - \Phi_{w1}(t, x_e), \tag{70}
$$

$$\omega_e = \omega_f - \Phi_{w2}\left(t, x_e, y_e, \theta_e\right), \tag{71}$$

where Φ_{w1} and Φ_{w2} are defined as in (46) and (65), respectively.

Now, consider the positive definite Lyapunov function candidate for the system (69):

$$\begin{aligned} W_1 &= V_3 + \frac{1}{2}v_e^2 + \frac{1}{2}\omega_e^2 \\ &= \frac{1}{2}x_e^2 + \frac{1}{2}y_e^2 + \frac{1}{2c_{w2}}\theta_e^2 + \frac{1}{2}z^2 + \frac{1}{2}v_e^2 + \frac{1}{2}\omega_e^2. \end{aligned} \tag{72}$$

Using (66), the time derivative of W_1 along the solutions of (69) satisfies:

$$\begin{aligned} \dot{W}_1 = &-x_e\sigma\left(x_e\right) - \frac{c_{w1}}{c_{w2}}\theta_e^2 + \frac{1}{c_{w2}}\theta_e z - \frac{1}{c_{w1}c_{w2}}z^2 \\ &+v_e\left[\frac{\gamma_{wa}K_G k_m}{\gamma_{wb}r_w R_M}u_1 - \frac{\gamma_{wa}K_G^2 v_c}{\gamma_{wb}r_w^2\cos\left(\phi\right)}\left(\frac{k_m k_E}{R_M} + J_{M1}\omega_f\tan\left(\phi\right)\right)\right. \\ &\left.-\frac{1}{\gamma_{wb}}\left(M_{FR} + \frac{J_w}{l_w}\omega_f\sec^2\left(\phi\right)v_c\right) - \dot{\Phi}_{w1}\right] \\ &+\omega_e\left(\frac{k_m}{J_{M4}K_G R_M}u_4 - \frac{k_m k_E\omega_f}{J_{M4}R_M} + \frac{\tau_{FR4}}{J_{M4}K_G} - \dot{\Phi}_{w2}\right), \end{aligned} \tag{73}$$

where:

$$\begin{aligned} \dot{\Phi}_{w1} = &\dot{v}_r + v_r^{\min}\left(1 - \tan^2\left(x_e\right)\right)y_e\tan\left(\phi\right)\left(\frac{v_r + \sigma\left(x_e\right)}{l_w}\right) \\ &+v_r^{\min}\left(1 - \tan^2\left(x_e\right)\right)\left(v_r\cos\left(\theta_e\right) - v_r - \sigma\left(x_e\right)\right), \end{aligned} \tag{74}$$

$$\begin{aligned} \dot{\Phi}_{w2} = &\frac{1}{l_w}\left[\left(\dot{\iota}_{w1}\sin\left(\phi\right) + \iota_{w1}\cos\left(\phi\right)\omega_f\right)\cos\left(\phi\right) - \iota_{w1}\sin^2\left(\phi\right)\omega_f\right] \\ &+\frac{\cos^2\left(\phi\right)}{v_c}\left[\dot{\iota}_{w2} + c_{w3}\dot{z} - \left(2\tan\left(\phi\right)\omega_f v_c + \dot{v}_c\right)\left(\iota_{w2} + c_{w3}z\right)\right], \end{aligned} \tag{75}$$

and:

$$\begin{aligned} \dot{\iota}_{w1} = &\dot{y}_e\tan\left(\phi\right) + \frac{y_e\omega_f}{\cos^2\left(\phi\right)} - c_{w2}\dot{v}_r\left(\alpha_{w1}x_e + \alpha_{w2}y_e\right) \\ &-c_{w2}v_r\left(\dot{\alpha}_{w1}x_e + \alpha_{w1}\dot{x}_e + \dot{\alpha}_{w2}y_e + \alpha_{w2}\dot{y}_e\right), \end{aligned} \tag{76}$$

$$
\begin{aligned}
i_{w2} = {} & \frac{\ddot{v}_r l_w + c_{w1}\dot{v}_r + c_{w2}v_r^2\dot{\alpha}_{w3}}{l_w}\tan(\phi_r) + c_{w2}\left(\ddot{v}_r\beta_w + \dot{v}_r\dot{\beta}_w\right) \\
& + \frac{2\dot{v}_r\omega_r + v_r\dot{\omega}_r + 2v_r\omega_r^2\tan(\phi_r)}{\cos^2(\phi_r)} + \frac{c_{w1}v_r\omega_r}{l_w\cos^2(\phi_r)} \\
& + \frac{c_{w2}v_r}{l_w}\left[2\dot{v}_r\tan(\phi_r) + \frac{v_r\omega_r}{\cos^2(\phi_r)}\right]\alpha_{w3}\frac{(v_r\cos(\theta_e) - v_c)\,\omega_f}{\cos^2(\phi)} \qquad (77) \\
& - c_{w2}\left[2v_r\dot{v}_r + \dot{v}_c v_r + v_c\dot{v}_r\right]\alpha_{w4} - c_{w2}\left(v_r^2 + v_c v_r\right)\dot{\alpha}_{w4} \\
& + \left(\dot{v}_r\cos(\theta_e) - v_r\sin(\theta_e)\,\dot{\theta}_e - \dot{v}_c\right)\tan(\phi),
\end{aligned}
$$

$$\alpha_{w4} = \frac{\cos(\theta_e) - 1}{\theta_e}, \qquad (78)$$

$$\dot{\alpha}_{w1} = \frac{\dot{\theta}_e}{\theta_e^3}\left[\theta_e\sin(\theta_e) + 2\cos(\theta_e) - 2\right], \qquad (79)$$

$$\dot{\alpha}_{w2} = \frac{\dot{\theta}_e}{\theta_e^3}\left[2\sin(\theta_e) - \theta_e\cos(\theta_e) - \theta_e\right], \qquad (80)$$

$$
\begin{aligned}
\dot{\alpha}_{w3} = {} & \frac{1 - \theta_e\sin(\theta_e) - \cos(\theta_e)}{\theta_e^2}\left(\dot{x}_e - 2\frac{\dot{\theta}_e}{\theta_e}x_e\right) \\
& + \frac{\theta_e\cos(\theta_e) - \sin(\theta_e)}{\theta_e^2}\left(\dot{y}_e - 2\frac{\dot{\theta}_e}{\theta_e}y_e\right) \qquad (81) \\
& - \frac{\dot{\theta}_e}{\theta_e}\left(x_e\cos(\theta_e) + y_e\sin(\theta_e)\right),
\end{aligned}
$$

$$\dot{\alpha}_{w4} = \frac{\dot{\theta}_e}{\theta_e^2}\left(1 - \theta_e\sin(\theta_e) - \cos(\theta_e)\right), \qquad (82)$$

$$\dot{\beta}_w = -\frac{\dot{\theta}_e}{\theta_e}\left(\beta_z + \delta_z\theta_e - y_e\right) + \frac{\cos(\theta_e) - 1}{\theta_e}\dot{x}_e + \frac{\sin(\theta_e)}{\theta_e}\dot{y}_e. \qquad (83)$$

We have let $\dot{W}_1$ be an explicit function of u_1 and u_4. Now, tracking control laws for u_1 and u_4 are selected to make $\dot{W}_1$ negative definite. A simple way to achieve this is taking:

$$u_1 = \frac{R_M K_G v_c}{\rho k_m \cos^2(\phi)} \left(\frac{k_m k_E}{R_M} + J_{M1} \omega_f \tan(\phi) \right)$$
$$+ \frac{1}{\gamma_{wa}} \left(M_{FR} + \frac{J_w}{l_w} \omega_f \sec^2(\phi) \, v_c \right) + \frac{\gamma_{wb}}{\gamma_{wa}} \left(\dot{\Phi}_{w1} - c_{w4} v_e \right), \tag{84}$$

$$u_4 = K_G k_E \omega_f + \frac{R_M}{k_m} \tau_{FR4} + \frac{J_w K_G R_M}{k_m} \left(\dot{\Phi}_{w2} - c_{w5} \omega_e \right). \tag{85}$$

With these control laws, the CLF derivative is:

$$\dot{W}_1 \leq -x_e \sigma(x_e) - \frac{c_{w1}}{2c_{w2}} \theta_e^2 - \frac{1}{2c_{w1}c_{w2}} z^2 - c_{w4} v_e^2 - c_{w5} \omega_e^2 \leq 0. \tag{86}$$

Therefore, if $c_{w4}, c_{w5} > 0$, the asymptotic stability that was originally obtained by the kinematic controller is preserved for the overall closed loop system.

4.3 Traction Force Distribution

A specific global traction force $\overrightarrow{F_{DG}}$ is required to support the velocity during the movement. Because of the proposed robot configuration, there is possibility to distribute this global traction force $\overrightarrow{F_{DG}}$ between the three wheels:

$$\overrightarrow{F_{DG}} = \overrightarrow{F_{D1}} + \overrightarrow{F_{D2}} + \overrightarrow{F_{D3}}. \tag{87}$$

Then, the objective in this subsection is to present a new control algorithm that distributes the global traction force between the wheels and prevents the chances of sliding. For a proper understanding of this algorithm, it is necessary to be familiar with static friction and sliding friction concepts.

If a torque is applied at a wheel that is not in contact with a surface, no translational movement is obtained, and subsequently, the wheel spins but it does not roll. If the wheel is in contact with a surface, the applied torque also tries to rotate the wheel. However, as the wheel is now in contact with a surface, a force of friction (static or sliding) appears between the outer edge of the wheel and the surface. This force, applied at the outer edge of the wheel, has a forward direction and generates a translation. On the other hand, the force of friction is limited. Then, if we start to increase the torque applied to the axle of the wheel, there is a limit torque value when the force of friction that appears at the outer edge of the wheel is equal to the maximum possible force of friction. If the torque applied to the axle of the wheel is smaller than the limit torque, a static friction force appears in the point of contact with the surface generating a translation without slipping. This static friction force is known as traction force. When the condition of rolling without slipping is fulfilled, the rotational movement is related with the translational movement as follows:

$$V = \dot{\varphi} r_w, \tag{88}$$

where V is the linear velocity in the center of the wheel, $\dot{\varphi}$ is the angular velocity and r_w is the radius of the wheel.

If the torque applied to the axle of the wheel is increased more than the limit torque, the force of friction cannot prevent the slipping and the wheel begins to move with rotation and sliding. Then, if there is sliding, $V \neq \dot{\varphi} r_w$.

Summarizing, if the torque applied to the wheel results in a force that is less than the resistive force of friction at the ground, the wheel will roll forward without slipping. If the torque is large enough to overcome the static force of friction, the wheel will slip. In this case, the wheel still moves forward, since the force of friction is now sliding friction. Nevertheless, note that the sliding force of friction is less than the static force. Thus, spinning the wheels is not an efficient way to move forward.

Taking into account that the environment of the desired application for our wheeled mobile robot has a small coefficient of static friction, it will be challenging to advance forward by rolling. Moreover, the control of the direction and velocity of the wheeled mobile robot will be a critical task. To overcome these difficulties, we propose a load torque distribution that is based on the idea that a load torque is proportional to a limit load torque on each wheel:

$$\frac{\tau_{ap1}}{T_{ap1}} = \frac{\tau_{ap2}}{T_{ap2}} = \frac{\tau_{ap3}}{T_{ap3}}, \tag{89}$$

where τ_{ap1}, τ_{ap2}, τ_{ap3} are the load torques applied at the wheels, and T_{ap1}, T_{ap2}, T_{ap3} are the limit load torques.

It was explained above that when the motor performs a torque greater than the maximum static friction force, the wheel slips and does not move forward efficiently. Then, the "maximum possible load torque" of each motor is connected with the maximum possible force of static friction and it is the torque magnitude from which the sliding conditions begin. If some normal force sensors are installed for each wheel, the real maximum load torque can be calculated using the equation:

$$Tr_i = N_i \mu r_w, \qquad \forall i, \quad i = 1 - 3, \tag{90}$$

where N is the normal force and μ is the coefficient of static friction. Because of the risk of operating too close to the real maximum load torque, a safety margin has been imposed for the wheeled mobile robot by defining a limit load torque T_{api} smaller than the real maximum load torque Tr_i.

Thus, the proposed tracking algorithm controls the traction motors load torques of the rear wheels in such a way that the magnitude of the load torque measured from the front wheel is used to minimize the following error function (see Fig. 4):

$$\tau_{ei} = \tau_{api} - \tau_{ap1} \cdot \frac{T_{api}}{T_{ap1}} \qquad \forall i, \quad i = 2 - 3, \tag{91}$$

where τ_{api} is the load torque on the wheel i, and T_{api} is the maximum possible load torque on the wheel i.

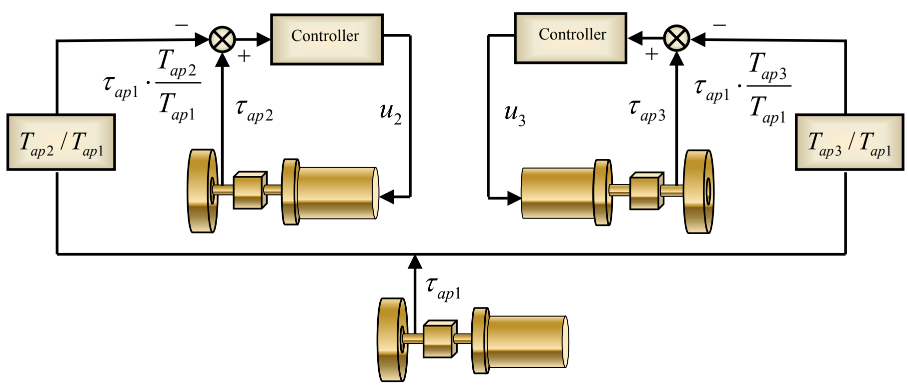


Fig. 4. Traction force distribution.

The ratio $\frac{T_{api}}{T_{ap1}}$ is defined as a traction force distribution coefficient λ_i. The magnitude of this coefficient λ_i, is related with the distribution of normal forces. Remember that in the subsection 4.2, λ_{w2} and λ_{w3} let us model the traction forces applied at the rear wheels as some percent of the traction force applied at the front wheel. Thus, it was viable to derive the feedback control law for u_1, but taking into account the effects of u_2 and u_3. Now, it is desirable to design control laws for u_2 and u_3 that guarantee asymptotic tracking of the reference load torques defined by the traction force distribution algorithm. Since the wheeled mobile robot motion is a quasi-static process, the load torques are approximately equal to the internal motor torques. In the same way, the internal motor torques are proportional to the armature currents. Therefore, in order to simplify the design of the control laws for u_2 and u_3 and facilitate the experimental evaluation, it is preferable to work with the currents I_1, I_2, and I_3, instead of the applied torques τ_{ap1}, τ_{ap2}, τ_{ap3}. The tracking errors are then given by:

$$e_2 = I_2 - \lambda_{w2} I_1, \tag{92}$$

$$e_3 = I_3 - \lambda_{w3} I_1. \tag{93}$$

Before to state the derivatives of (92) and (93), it is convenient to find out the relationship between the linear velocity of the reference point, v_c, and the angular velocities of the the rear wheels' rotors, $\dot{\varphi}_2$ and $\dot{\varphi}_3$. The relationship between the linear velocities projections of the center of the rear wheels and the angular velocities of their motors are given by:

$$\dot{x}_i = \frac{r_w}{K_G}\dot{\varphi}_i \cos(\theta), \qquad \forall i, \quad i = 2, 3 \tag{94}$$

$$\dot{y}_i = \frac{r_w}{K_G}\dot{\varphi}_i \sin(\theta), \qquad \forall i, \quad i = 2, 3. \tag{95}$$

Combining equations (5), (6), (15), (16), (94) and (95), the next equalities are obtained:

$$\begin{aligned} v_c \cos(\theta) - \frac{d_a}{2}\dot{\theta}\cos(\theta) &= \frac{r_w}{K_G}\dot{\varphi}_2 \cos(\theta), \\ v_c \sin(\theta) - \frac{d_a}{2}\dot{\theta}\sin(\theta) &= \frac{r_w}{K_G}\dot{\varphi}_2 \sin(\theta), \end{aligned} \tag{96}$$

$$\begin{aligned} \cos(\theta) + \frac{d_a}{2}\dot{\theta}\cos(\theta) &= \frac{r_w}{K_G}\dot{\varphi}_2 \cos(\theta), \\ v_c \sin(\theta) + \frac{d_a}{2}\dot{\theta}\sin(\theta) &= \frac{r_w}{K_G}\dot{\varphi}_2 \sin(\theta). \end{aligned} \tag{97}$$

From these equalities, it can be found that:

$$\dot{\varphi}_2 = \frac{K_G}{r_w}\left(1 - \frac{d_a \tan(\phi)}{2l_w}\right) v_c, \tag{98}$$

$$\dot{\varphi}_3 = \frac{K_G}{r_w}\left(1 + \frac{d_a \tan(\phi)}{2l_w}\right) v_c. \tag{99}$$

Now, the derivatives of (92) and (93) can be stated as follows:

$$\dot{e}_2 = \frac{1}{l_w}\left[u_2 - \lambda_{w2}u_1 - R_M e_2 + \frac{k_E K_G v_c}{r_w}\left(\frac{\lambda_{w2}}{\cos(\phi)} - 1 + \zeta\right)\right] \tag{100}$$

$$\dot{e}_3 = \frac{1}{l_w}\left[u_3 - \lambda_{w3}u_1 - R_M e_3 + \frac{k_E K_G v_c}{r_w}\left(\frac{\lambda_{w3}}{\cos(\phi)} - 1 - \zeta\right)\right] \tag{101}$$

where

$$\zeta = \frac{d_a \tan(\phi)}{2 l_w} \tag{102}$$

Taking:

$$\begin{aligned} W_2 &= W_1 + \frac{l_w}{2}\left(e_2^2 + e_3^2\right) \\ &= \frac{1}{2}\left(x_e^2 + y_e^2 + \frac{1}{c_{w2}}\theta_e^2 + z^2 + v_e^2 + \omega_e^2 + l_w e_2^2 + l_w e_3^2\right) \end{aligned} \tag{103}$$

as a positive definite proper Lyapunov function candidate for the system (69, 100, 101), and choosing:

$$u_2 = \lambda_{w2}u_1 - \frac{k_E K_G v_c}{r_w}\left(\frac{\lambda_{w2}}{\cos(\phi)} - 1 + \frac{d_a \tan(\phi)}{2 l_w}\right) - c_{w6}e_2, \tag{104}$$

$$u_3 = \lambda_{w3}u_1 - \frac{k_E K_G v_c}{r_w}\left(\frac{\lambda_{w3}}{\cos(\phi)} - 1 - \frac{d_a \tan(\phi)}{2 l_w}\right) - c_{w7}e_3, \tag{105}$$

yields:

$$\begin{aligned} \dot{W}_2 \leq & -x_e \sigma(x_e) - \frac{c_{w1}}{2c_{w2}}\theta_e^2 - \frac{1}{2c_{w1}c_{w2}}z^2 - c_{w4}v_e^2 - c_{w5}\omega_e^2 \\ & - (R_M + c_{w6})\, e_2^2 - (R_M + c_{w7})\, e_3^2 \qquad \leq 0, \end{aligned} \tag{106}$$

where $c_{w6}, c_{w7} > 0$. This implies asymptotical stability according to Lyapunov stability theorem.

Remark 3. Notice that the main tasks of the proposed traction force distribution algorithm are to define the proper reference load torque for each rear wheel and to guarantee their asymptotic tracking.

Remark 4. To prevent an excess of traction force that can produce slippage during the robot movement, the control algorithm verifies that the traction motor load torque applied to each wheel does not exceed its corresponding limit load torque T_{api}. However, the situation is very peculiar.

While the traction motor load torques τ_{api} remain within certain limits, the voltages applied to the traction motors can be modified by the velocity control for front wheel and by the force control for rear wheels. However, when one of the traction motor load torques τ_{api} starts to be equal to the limit magnitude, it is necessary to hold all motor voltages neglecting the conditions established by the velocity control and force control for front and rear wheels, respectively. It is important to note, that even if control system starts to increase voltage, it will not lead to a velocity increasing but only to the occurrence of one or several wheels slippage. Therefore, this voltage has to be held until the moment in which the voltage needed to support the velocity is smaller than the needed voltage to support the limit load torque. In assumption 4.1 we assumed that the forward reference velocity is bounded such that $0 < v_r^{\min} \leq v_r(t) \leq v_r^{\max}$. Then, to avoid the loss of stability in the aforementioned situation, $v_r^{\max}$ has to be selected according to the T_{api} values.

5 Experimental Results

To validate the proposed method, a real time implementation of the control system was developed and numerous experimental trials were realized using a prototype of the mobile robot (see Fig. 5). The prototype has three identical wheels which are controlled by three identical DC motors. The wheels have a radius of $0.05m$. The front wheel is also steerable by an additional DC motor. Table 1 shows the values of the motors' parame-

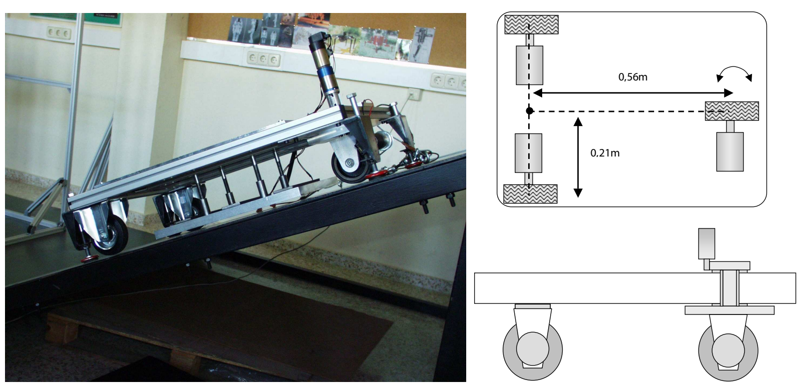
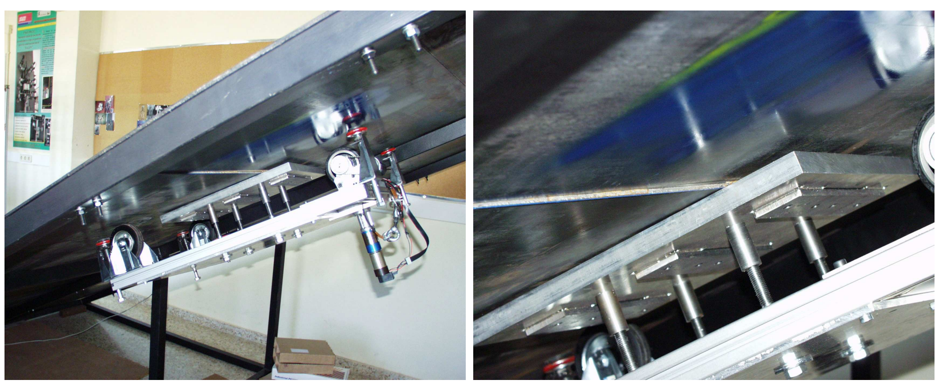


Fig. 5. Wheeled mobile robot prototype.

ters. The size of the mobile robot prototype is $0.65m$ length and $0.5m$ width. Its mass is $36Kg$ in total and its moment of inertia around the center of mass is $2Kg/m^2$. There are six permanent magnets of vanadium $(50 \times 50 \times 50)\, mm$ installed in a rigid plate of aluminium that let the robot climb along ferromagnetic surfaces (see Fig. 6). This plate is mounted in the mobile robot's platform in such a way that the distance between the magnets and the ferromagnetic surface can be varied in parallel. The values of the remaining system parameters are given in Table 2. The gains used for the state-feedback controllers (84), (104), (105) and (85) are: $c_1 = c_2 = 0.56$, $c_3 = 7$, $c_4 = 10$, $c_5 = 3$, $c_6 = c_7 = 2$.

Table 1. Parameters of the DC motors.

Parameters	Values
R_M	$0.6\,\Omega$
k_m	$0.0333\ Nm/A$
k_E	$0.0425\ Vrad/s$
K_G	246
$J_{M1} = J_{M2} = J_{M3} = J_{M4}$	$0.000011\ Kg/m^2$



(a) (b)

Fig. 6. (a) Wheeled mobile robot prototype climbing a ferromagnetic surface. (b) Permanent magnets.

Table 2. Parameters of the wheeled mobile robot prototype.

Parameters	Values
m_w	$36\,Kg$
J_w	$2\ Kg/m^2$
d_a	$0.21\ m$
l_w	$0.56\ m$
r_w	$0.05\,m$
μ	0.1

In one experiment, the robot was programmed to move upon a horizontal flat surface with $v_r(t) = 0.0262m/s$, $\phi_r(t) = 0°$, and in such way that the traction motor load torques of the rear wheels were equal to the measured load torque from the front wheel. This means a traction force distribution of 1 : 1 : 1. Fig. 7(a) and Fig. 7(b) show the time evolution of x_c and y_c respectively. Fig. 7(c) and Fig. 7(d) illustrate the time evolution of θ and ϕ respectively. Fig. 7(e) displays the behavior of the velocity while Fig. 7(f) shows the armature currents of the traction motors. Red dotted lines represent the desired values and blue solid lines represent the actual values. In this case the tracking performance is so satisfactory that it is difficult to distinguish the controlled signals from the references ones.

In a second experiment, the robot was programmed to move in a circular trajectory upon an inclined plane with $v_r(t) = 0.0305m/s$, $\phi_r(t) = 45°$ and with a traction force distribution of 1 : 1 : 1. Additionally, the robot had to move over two obstacles of about 1cm. Fig. 8 shows the same set of curves that were presented above. It is interesting to remark that all the armature currents are practically equal while the robot is turning. Most of the state variables exhibit an almost perfect tracking of their reference values, excepting ϕ. Nevertheless, its tracking error quickly drops to zero.

In the next experiment the robot was programmed to move upon an inclined plane with $v_r(t) = 0.0436m/s$ and with a traction force distribution of 1 : 0.9 : 0.7. Additionally, the robot had to move over two obstacles. Fig. 9(a) and Fig. 9(b) show the time evolution of x_c and y_c respectively. Fig. 9(c) and Fig. 9(d) illustrate the time evolution of θ and ϕ respectively. Fig. 9(e) shows the desired velocity and the measured velocity on the reference point R_p. Fig. 9(f) shows the armature currents of the traction motors. Note the good tracking performance despite the traction force distribution and the presence of obstacles.

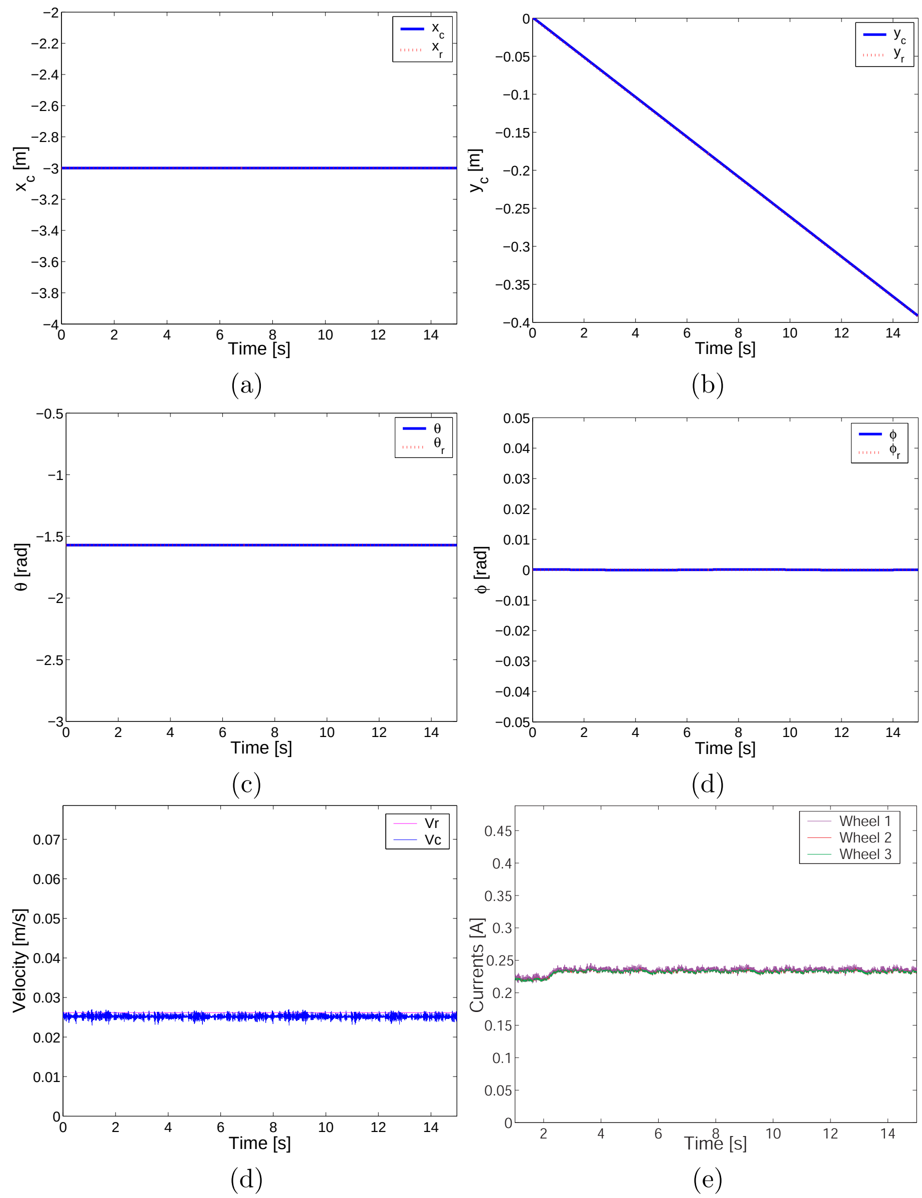
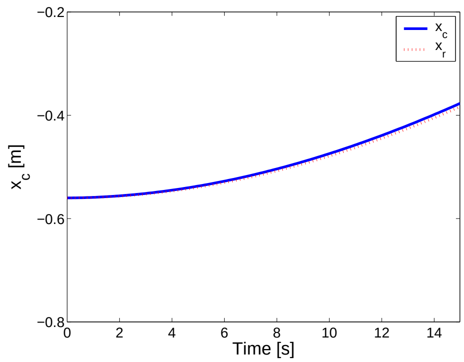
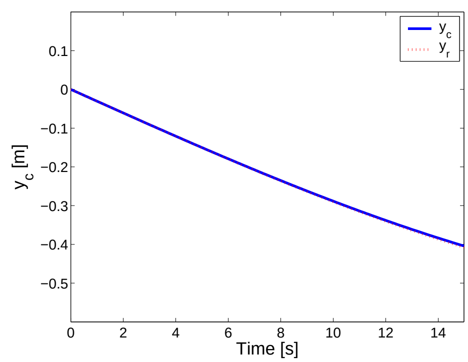


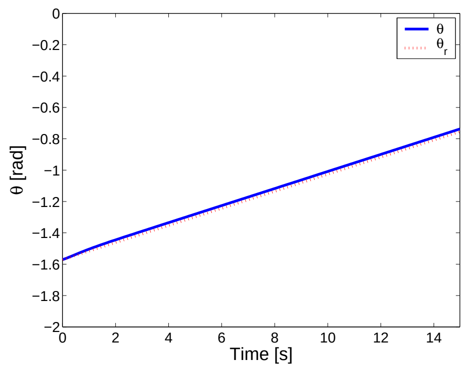
Fig. 7. Experimental tracking results with $v_r\left(t\right) = 0.0262m/s$, $\phi_r\left(t\right) = 0°$ and $I_2 = I_3 = I_1$.



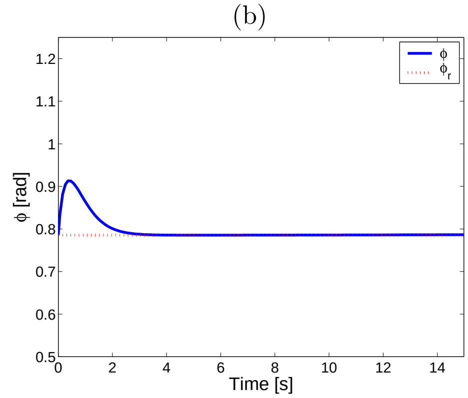
(a)



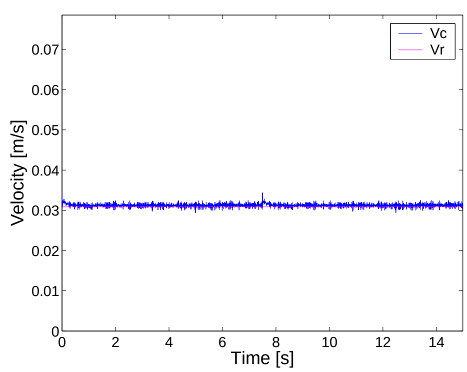
(b)



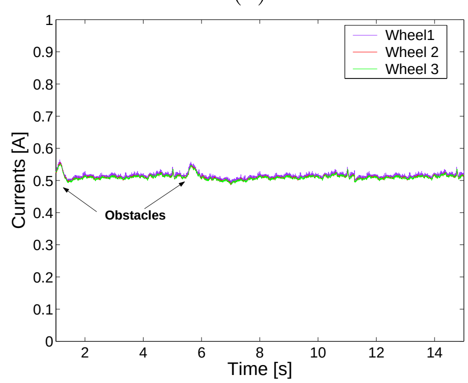
(c)



(d)



(e)



(f)

Fig. 8. Experimental tracking results with $v_r\,(t) = 0.0305m/s$, $\phi_r\,(t) = 45°$ and $I_2 = I_3 = I_1$.

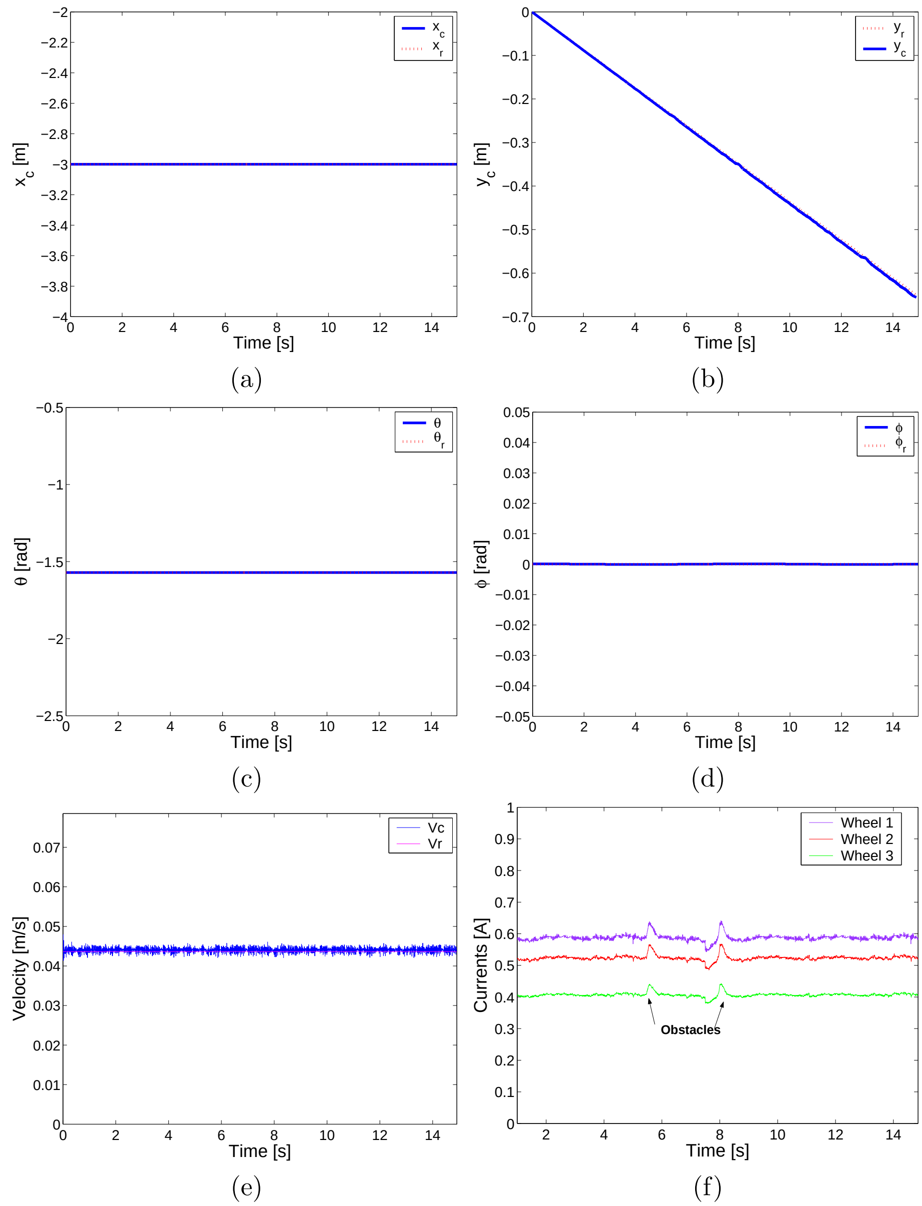


Fig. 9. Experimental tracking results with $v_r(t) = 0.0436m/s$, $\phi_r(t) = 0°$, $I_2 = 0.9I_1$ and $I_3 = 0.7I_1$.

In the next experiment, the robot was programmed to move up along a vertical wall with $v_r(t) = 0.0436m/s$ and with a traction force distribution of $1 : 2.5 : 2.5$. Additionally, the robot was charging a load of $10kg$. Fig. 10(a) and Fig. 10(b) show the time evolution of x_c and y_c respectively. Fig. 10(c) and Fig. 10(d) illustrate the time evolution of θ and ϕ respectively. Fig. 10(e) shows the desired velocity, v_r, the measured velocity on the reference point, v_c, and the measured linear velocities in the center of each wheel. Fig. 10(f) shows the armature currents of the traction motors.

In the next experiment, the robot was programmed again to move up along a vertical wall with $v_r(t) = 0.0436m/s$, with a traction force distribution of $1 : 2.5 : 2.5$, and charging a load of $10kg$. Nevertheless, this time the robot had to move over two obstacles of about 1cm. Fig. 11(a) and Fig. 11(b) show the time evolution of x_c and y_c respectively. Fig. 11(c) shows the desired velocity, v_r, the measured velocity on the reference point, v_c, and the measured linear velocities in the center of each wheel. Fig. 11(d) shows the armature currents of the traction motors. In this case, a small effect of the obstacles is appraisable in the state variable y_c. However the tracking performance is satisfactory.

In the next experiment the robot was programmed to turn 25° to the left upon a vertical ferromagnetic wall, with $v_r(t) = 0.04m/s$ and with a traction force distribution of $1 : 2.5 : 1$. Fig. 12(a) and Fig. 12(b) show the time evolution of x_c and y_c respectively. Fig. 12(c) and Fig. 12(d) illustrate the time evolution of θ and ϕ respectively. Fig. 12(e) shows the desired velocity, v_r, the measured velocity on the reference point, v_c, and the velocity on each wheel. Fig. 12(f) displays the armature currents on the traction motors.

Fig. 13 shows the experimental results obtained when the wheeled mobile robot is programmed to move upon a horizontal surface describing a circular trajectory of radius $1.2009m$ with $v_r(t) = 0.04m/s$, $\phi_r(t) = 0°$, $I_2 = 2.5I_1$ and $I_3 = I_1$. The reference virtual mobile robot sets out from point $(x_r(0), y_r(0), \theta_r(0)) = \left(-1.2009, 0, -\frac{\pi}{2}\right)$. The actual robot starts at point $(x_c(0), y_c(0), \theta(0)) = \left(-0.80, 0, -\frac{\pi}{2}\right)$. This means that the initial error is $(0.4009, 0, 0)$. Note that there is no path planning involved. It is worth noticing that the tracking errors drop to zero, and therefore, the state variables converge to the reference values.

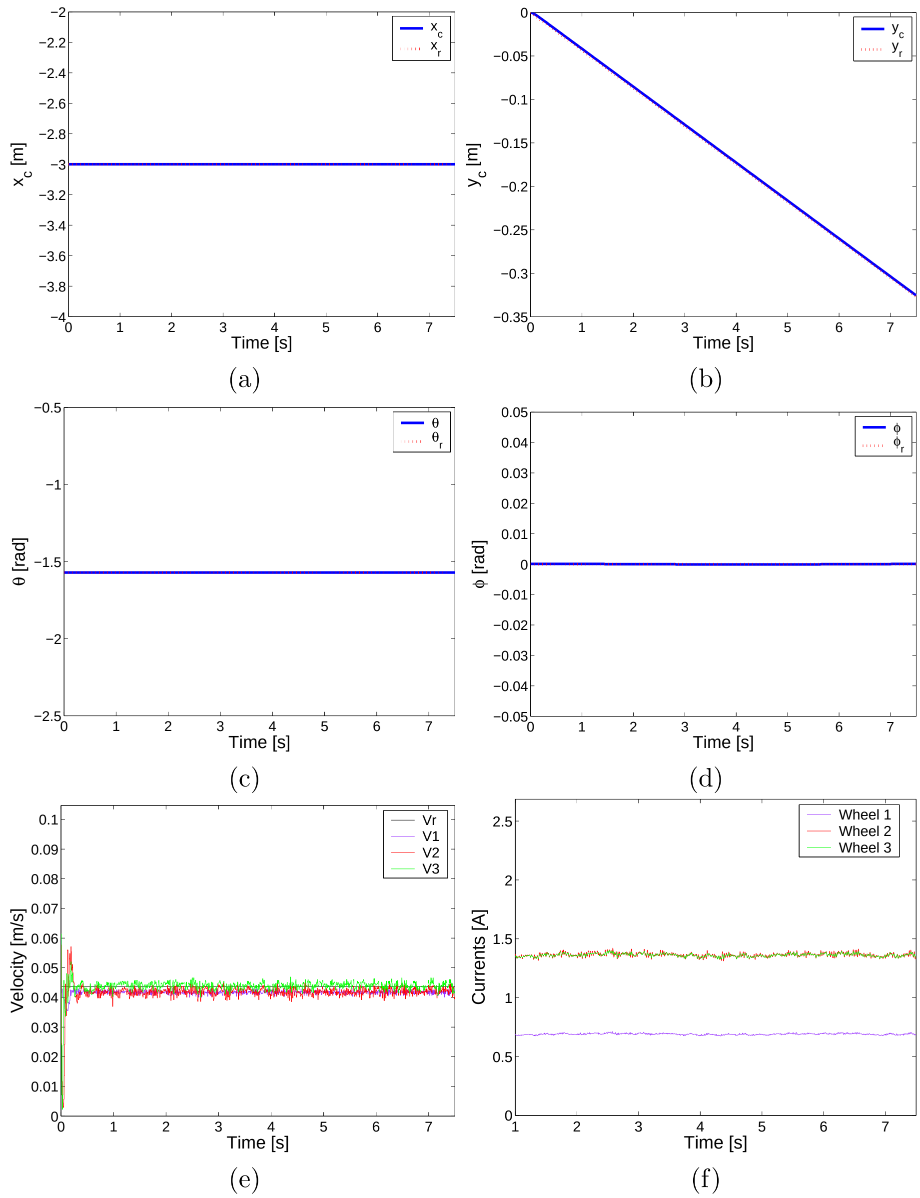


Fig. 10. Experimental tracking results with $v_r\,(t) = 0.0436m/s$, $\phi_r\,(t) = 0°$ and $I_2 = I_3 = 2.5I_1$.

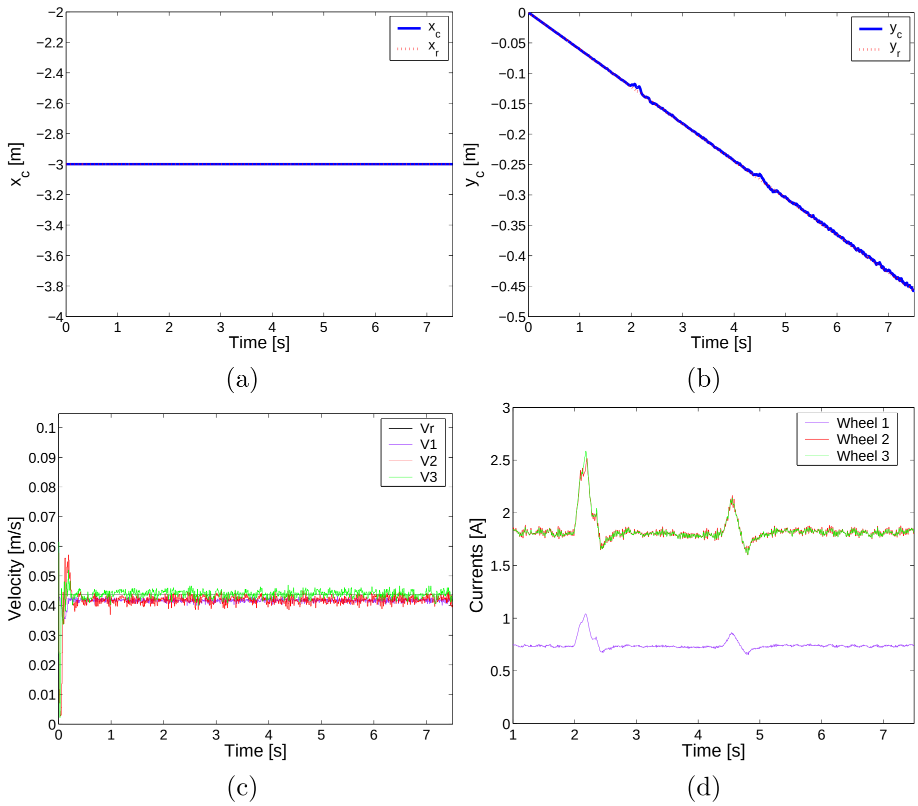


Fig. 11. Experimental tracking results with $v_r(t) = 0.0436m/s$, $\phi_r(t) = 0°$, $I_2 = I_3 = 2.5I_1$ and the presence of two obstacles.

Lastly, Fig. 14 shows the dynamic tracking performance of the wheeled mobile robot in the $x - y$ plane for different initial conditions. The robot was programmed to move upon a horizontal surface with $v_r(t) = 0.04m/s$, $\phi_r(t) = 25°$ and with a a traction force distribution of $1 : 1 : 1$. Initially the robot is at $(x_r(0), y_r(0), \theta_r(0)) = \left(-1.2009, 0, -\frac{\pi}{2}\right)$. In Fig. 14(a) the robot starts at point $(x_c(0), y_c(0), \theta(0)) = (-0.80, -0.2, -\frac{\pi}{2})$. This means that the initial error is $(0.4009, -0.2, 0)$. In Fig. 14(b) the robot starts at point $(x_c(0), y_c(0), \theta(0)) = \left(-0.80, 0.5, -\frac{\pi}{2}\right)$. Therefore, the initial error is $(0.4009, 0.5, 0)$. In Fig. 14(c) the robot starts at point $(x_c(0), y_c(0), \theta(0)) = \left(-1.5, 0.3, -\frac{\pi}{2}\right)$. This means that the initial error is $(-0.2991, 0.3, 0)$. In Fig. 14(d) the robot starts at point $(x_c(0), y_c(0), \theta(0)) = \left(-1.2009, -0.5, -\frac{\pi}{2}\right)$.

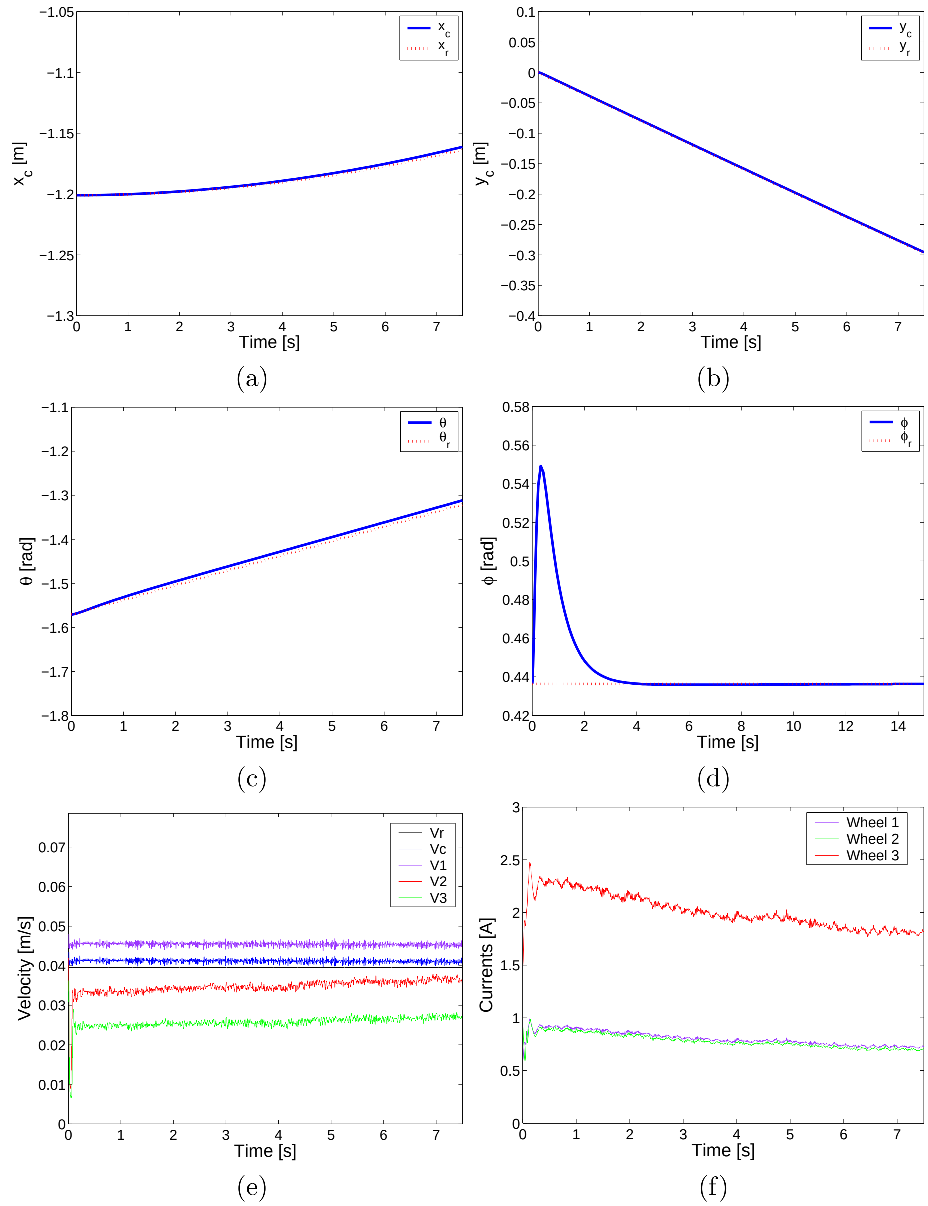


Fig. 12. Experimental tracking results with $v_r(t) = 0.04m/s$, $\phi_r(t) = 25°$, $I_2 = 2.5I_1$ and $I_3 = I_1$.

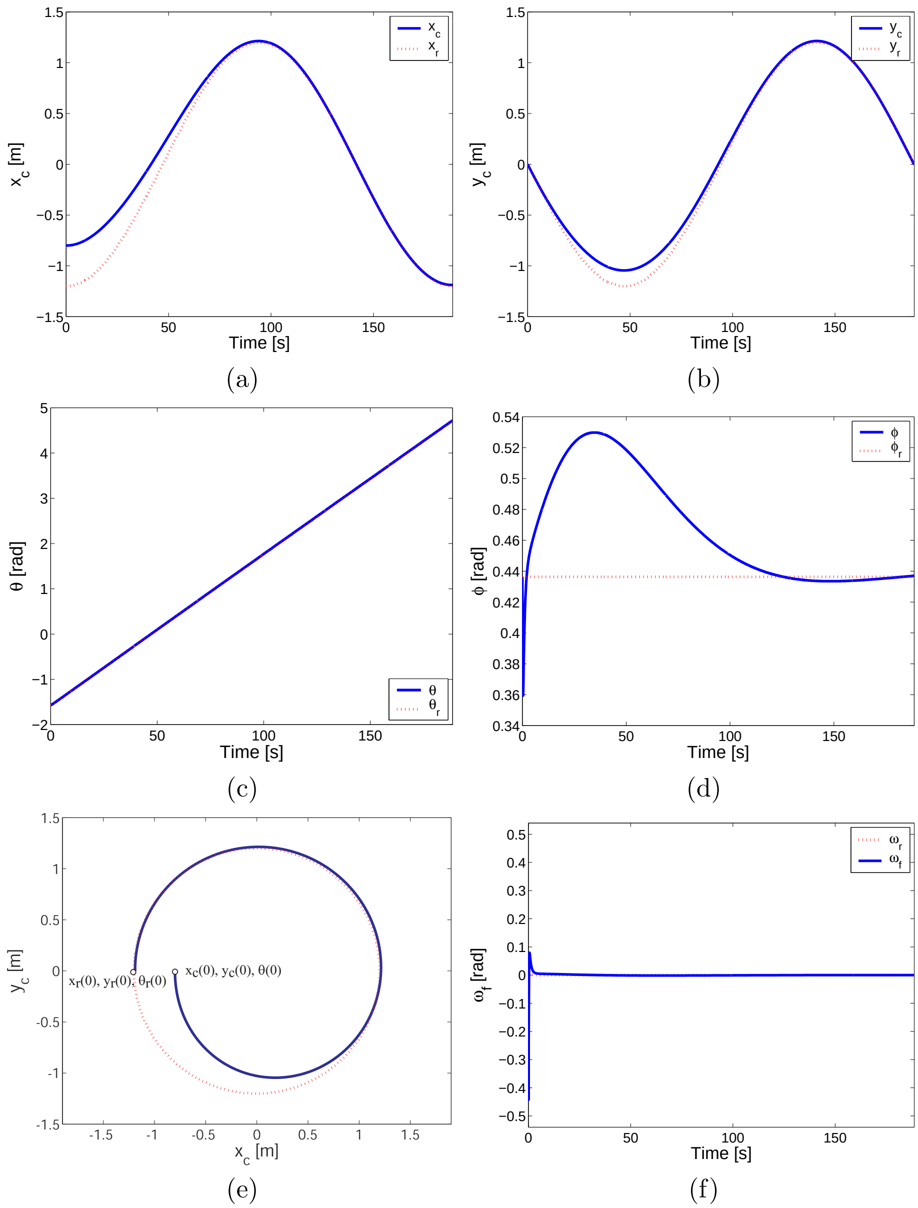


Fig. 13. Experimental tracking results with $v_r\,(t) = 0.04m/s$, $\phi_r\,(t) = 25°$, $I_2 = 2.5I_1$ and $I_3 = I_1$.

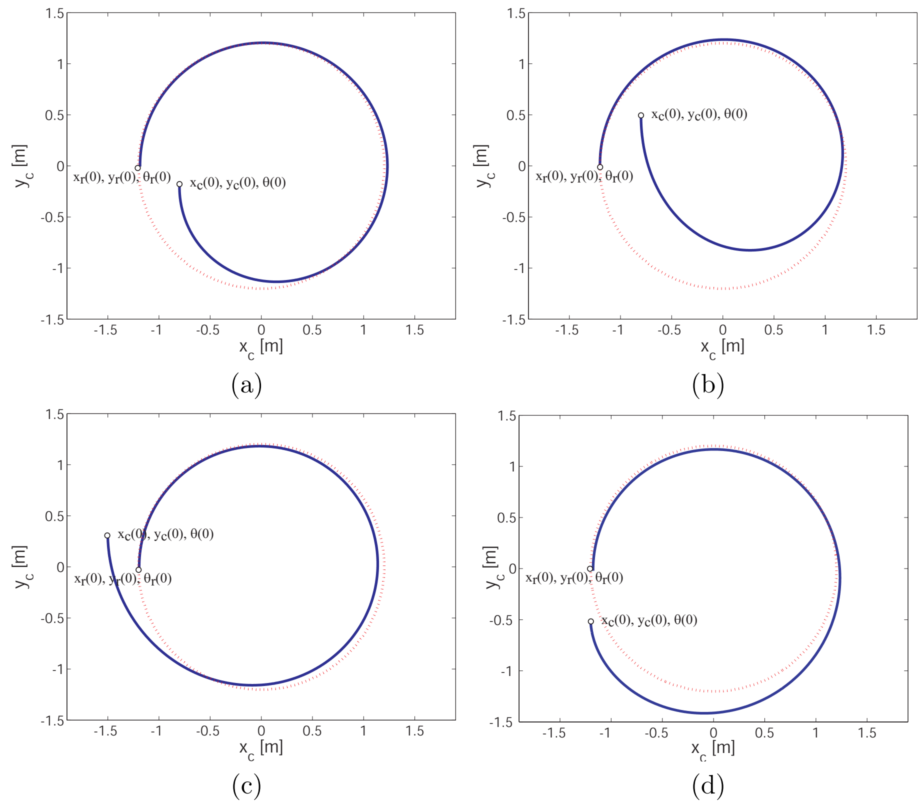


Fig. 14. Experimental tracking results with $v_r\,(t) = 0.04m/s$, $\phi_r\,(t) = 25°$ and $I_2 = I_3 = I_1$ for different initial conditions.

Then, the initial error is $(0, -0.5, 0)$. All the presented experimental results demonstrate that the control objectives were achieved.

Once the feasibility and satisfactory performance of the control laws were experimentally confirmed, we proceeded to design and manufacture a second prototype. This new prototype, displayed in Fig. 15-16 was equipped with the proper tools for the cleaning tasks (see Fig. 17).

A Supervisor Station comprised all the required elements for the remote control and supervision of all the AURORA functions. It was built around a COMPAQ workstation. All software was written in Visual C++, with an important number of libraries (Armada *et al.*, 2002). Fig. 18(a) illustrates some windows of the Human Machine Interface. A simulation environment was also implemented to allow the operator of the Aurora system to plan, verify, supervise and follow-up shipside cleaning tasks. A

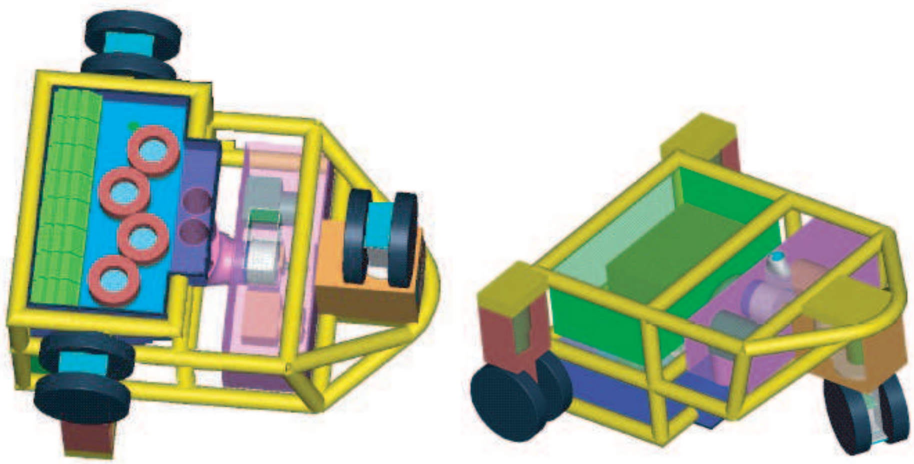


Fig. 15. Cleaning robot design.

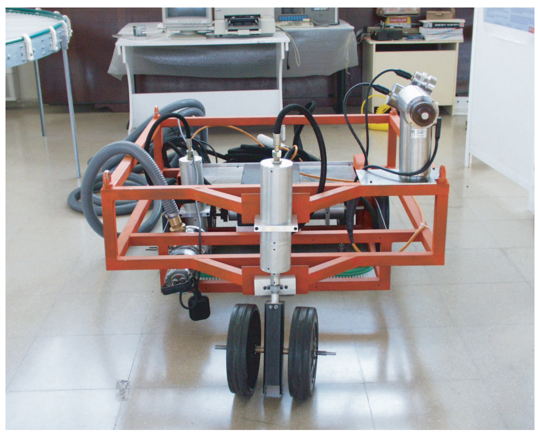


Fig. 16. Second prototype of the cleaning robot.

virtual model of the ship and cleaning device was used to visualize all stages in the process (see Fig. 18(b)).

6 Conclusions

AURORA, the European Project N° : GRD1 − 1999 − 11153 and GRD3 − 2001 − 60056, Contract N° : G3RD − CT − 2000 − 00246, was aiming to

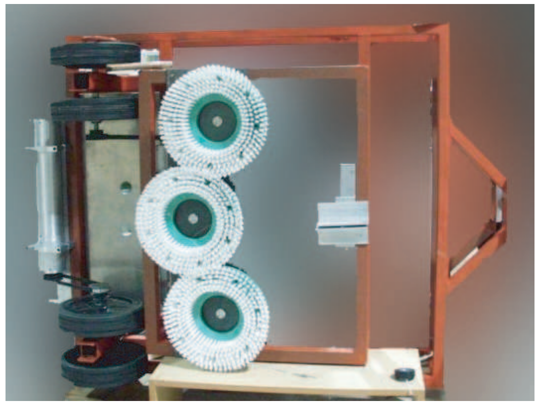
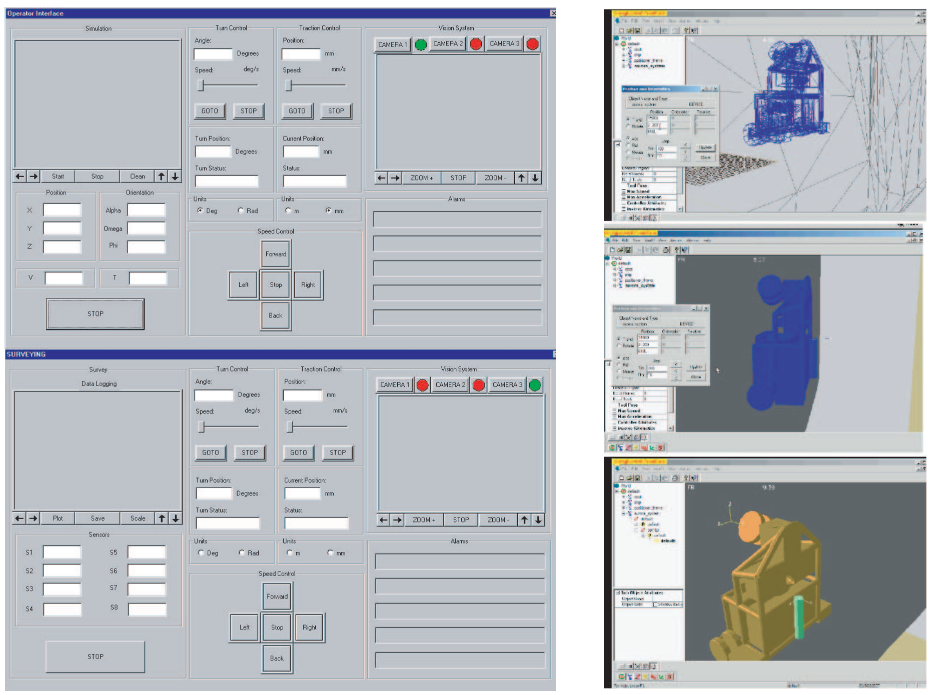


Fig. 17. Cleaning tools of the robot.



(a) (b)

Fig. 18. (a) User interfaces for automatic operation. (b) AURORA simulator.

provide an eco-efficient, economical, practical and safe solution to the underwater ship hull cleaning of sea adherences. The main technical achievement of the project was the design and control of a robot that is able to clean sea adherences from the outside surface of a ship (the ship hull surface mostly exposed to the sea water). Thus, the solution implies a non-classically actuated robot's configuration that increases the maximum possible global traction force and a controller that tracks the reference position and the reference velocity, and at the same time, distributes the traction force between the wheels in such a way that their sliding is avoided. This control method is composed of three steps. Firstly, feedback velocity control inputs are designed for the kinematic system to solve the nonholonomic problem and to make the position error asymptotically stable. Secondly, state-feedback controllers are derived such that the mobile robot's velocities converge to the given velocity inputs. Thirdly, a traction force distribution algorithm that calculates the proper reference signals for each rear wheel is included and the feedback tracking control laws are finally completed. The effectiveness and applicability of the proposed methodology have been confirmed by experimental results. In every situation, it was possible to distribute the traction forces between the wheels while the required position and velocity were supported. A significant increasing in the maximum possible traction force without sliding of the wheels was achieved. Therefore, the proposed configuration demonstrate to be quite useful for surfaces with small coefficient of static friction and in situations where a big resistance of the working process is present.

Acknowledgements

The authors would like to acknowledge the support of the European Commission, who funded this research in the frame of Competitive and Sustainable Growth Programme Project GRD1-1999-11153 "Auxiliary Climbing Robot for Underwater Ship Hull Cleaning of Sea Adherence and Surveying - AURORA".).

The authors would also like to acknowledge the financial support from Ministry of Science and Technology of Spain (Project "Theory of optimal dual drives for automation and robotics") and from Ministry of Culture and Education of Spain (Fellowship F.P.U.). Also authors would like to extend the acknowledgement to the CICYT (Ministry of Education) for the funding of AUTOMAR Network and to Consejería de Educación (Comunidad de Madrid) for funding RoboCity2030 (S-0505/dpi-0176) project.

Last but not least, authors would like to extend the acknowledgement to all the partners of the AURORA Project: two ship-repairing yards, T. Kalogeridis&Co. Inc. and Unión Naval de Barcelona, Algosystems S.A., the Division of Robotics, Department of Mechanical Engineering, from Lund University, SAIND, manufacturer and vendor of equipment for shipyards, and Riga Technical University.

References

Aguiar, A. P., Atassi, A. N., and Pascoal, A. M. 2000. Regulation of a nonholonomic dynamic wheeled mobile robot with parametric modeling uncertainty using Lyapunov functions. In Proceedings of 39^{th} IEEE Conference on Decision and Control, Sydney, Australia.

Akinfiev, T., Armada, M., and Fernández, R. 2001. Método de control de vehículos (Vehicle control method). Spanish Patent. Publication Number: 2 187 375.

Armada, M., González de Santos, P., Jiménez, M. A., Estremera, J., García, E., Prieto, M., and Fernández, R. 2002. Human-machine interfaces for teleoperation of climbing and walking robots. In Proceedings of the International Advanced Robotic Program, International Workshop on Human Robot Interfaces. Technologies and Applications, pages 109–117, Frascati, Roma.

Bestaoui, Y. 2000. An optimal velocity generation of a rear wheel drive tricycle along a specified path. In Proceedings of the American Control Conference, Chicago, Illinois.

Everett, H. R. 1995. Sensors for Mobile Robots: Theory and Application. AK Peters, Ltd.

Fernández, R., Akinfiev, T., and Armada, M. 2002. Control algorithms for an underwater climbing robot. In P. Bidaud and F. B. Amar, editors, Proceedings of the 5^{th} International Conference on Climbing and Walking Robots and their supporting Technologies., pages 985–992, London, UK. Professional Engineering Publishing Limited.

Fukao, T., Nakagawa, H., and Adachi, N. 2000. Adaptive tracking control of a nonholonomic mobile robot. IEEE Transactions on Robotics and Automation, **16**(5): 609–615.

Ge, S. S., Wang, J., Lee, T. H., and Zhou, G. Y. 2001. Adaptive robust stabilization of dynamic nonholonomic chained systems. Journal of Robotic Systems, **18**(3): 119–133.

Kanayama, Y., Kimura, Y., Miyazaki, F., and Noguchi, T. 1990. A stable tracking control scheme for an autonomous mobile robot. In Proceedings of the IEEE International Conference on Robotics and Automation, pages 384–389, Cincinnati, Ohio.

Lefer, E. and Nijmeijer, H. 1999. Adaptive tracking control of nonholonomic systems: an example. In Proceedings of the 38^{th} IEEE Conference on Decision and Control, pages 2094–2099, Phoenix, Arizona, USA.

CHAPTER 9

Constructing photo-mosaics to assist UUV navigation and station-keeping

R. GARCIA, X. CUFI, P. RIDAO and M. CARRERAS

Computer Vision and Robotics Group, Institute of Informatics and Applications, University of Girona, Spain,
{rafa,xcuf,pere,marcc}@eia.udg.es

This paper describes a system to build photo-mosaics generated from the images of a downward-looking camera. An Unmanned Underwater Vehicle (UUV) carries the camera, and the images of the ocean floor are used to construct a visual mosaic of the surveyed area. As the photo-mosaic is being constructed, it is simultaneously used as a visual map for vehicle positioning. The proposed system is divided into two modules: the mosaic *supervisor* and the mosaic *engine*. The nature of the motion performed by the vehicle is taken into account by the *supervisor*, which takes the pertinent actions necessary to assist the *engine* in the image registration phase. These actions consist in deciding which two images are best suited to be passed to the mosaic *engine* to better estimate the motion of the vehicle. The two images are then transferred to the *engine* along with an "a priori" estimate of the vehicle motion. This estimation is obtained from the data fusion of the available on-board sensors. Image registration is carried out by detecting the most prominent features in one of the images and finding their correspondences in the other through textural-aided correlation. A planar transformation then relates the images to a common mosaic frame enabling the construction of a composite image with the frames of the sequence which have been selected by the mosaic *supervisor*. Finally, a procedure to obtain the 3D motion of the vehicle is described. Keywords: Photo-mosaicking, Underwater Imaging, AUV Navigation.

1 Introduction

Over the past few years, visual mosaics have greatly advanced as a tool for undersea exploration and navigation. Unmanned Underwater Vehicles (UUVs), be autonomous or teleoperated, frequently incorporate video cameras to provide visual information to the scientists or robot operators. These cameras can be arranged in such a way that they look downwards to the ocean floor. This enables the construction of photo-mosaics of the underwater terrain. Two main applications are derived from these visual mosaics. On the one hand, they can be used by marine biologists and oceanic engineers to gain a global perspective of the site of interest (Singh, 1998; Eustice, 2000). In this case, no real-time processing is necessary since off-line construction of the mosaic should be enough, providing that the mosaic covers the surveyed area without any visual gaps. On the other hand, a down-looking camera provides rich visual information, under favorable visibility conditions, which can be used for vehicle localization, station-keeping, mapping and navigation. Several strategies have been presented in the literature to recover the vehicle motion by means of photo-mosaics.

A straight forward solution to construct a visual mosaic is based on processing every pair of consecutive images of the sequence, obtaining the incremental motion parameters which describe the motion of the camera with respect to its previous position (Gracias, 1998; Garcia, 2001a). These incremental measures can then be integrated to obtain global position estimates. Unfortunately, every incremental measure introduces a small bias which accumulates over the sequence.

Some alternatives have been presented to reduce the accumulated drift in the construction of the mosaic. Marks *et al.* (Marks, 1994b; Marks, 1995) proposed to add a new image to the photo-mosaic only if its overlapping region with the previously added image was smaller than a given threshold. This methodology leads to a smaller amount of images composing the mosaic in relation to the mapped area. Therefore, less drift is accumulated, but camera rotation and scaling should be small to enable motion estimation between both images.

A different solution was adopted by Gracias and Santos-Victor by first applying their consecutive-image strategy (Gracias, 1998; 2000), and then reducing the accumulated error by detecting areas of image overlap in non-consecutive images (Gracias, 2001). Next, point correspondences are searched in these images, and a new refined motion estimation is obtained. Once the whole photo-mosaic has been constructed, a global non-linear optimization is performed to further reduce image alignment errors using a more general motion model. The only drawback of this technique would be its high computational cost.

Negahdaripour *et al.* proposed in (Negahdaripour, 2002) to use a constant parameter L governing the actualization rate of the mosaic. In this way, the mosaic is only updated with a new image every L frames. When an image is selected to actualize the mosaic, a first estimation of the location of the image in the mosaic is computed through the registration of this image with the previous one. Then, an image is extracted from the mosaic in the estimated location and refined motion estimation is computed, thus reducing the accumulated error.

Trucco *et al.* (Trucco, 2000a; Trucco, 2000b; Lots, 2001) proposed another strategy which consisted in selecting a base image, and then estimating the motion of some image features over the sequence for as long as possible. When the tracked features are out of the field of vision of the camera, a new base image and new features have to be selected. Again, sources of error are minimized by detecting motion between non-consecutive overlapping images.

When performing a mission, the submersible follows an arbitrary path, which, may occasionally cross over itself. In this situation new information is also available, and the system can readjust the error margin associated to the position of the vehicle. Fleischer *et al.* (Fleischer, 1996; Fleischer, 1997) proposed to exploit the additional information gained when the vehicle path crosses itself by applying an iterative *smoother-follower* filter to optimally realign the chain of images composing the mosaic in the looping path.

Finally, Roman and Singh presented a strategy which involves the use of high-resolution acoustic navigation data for error estimation in the construction of photo-mosaics (Roman, 2001; Whitcomb, 1999). Although this work aimed to illustrate the existing distortions in apparently correct photo-mosaics, it introduces the subject of considering sensors other than vision to obtain a new source of measurement, which could later be used to improve image alignment.

Some of the previously described techniques (*e.g.* (Negahdaripour, 2002; Gracias, 2000; Fleischer, 1996)) implicitly exploit the advantages of *Concurrent Mapping and Localization* (CML) strategies (Smith, 1997). When applying a CML methodology, the constructed photo-mosaic provides supplementary information about the vehicle motion in addition to the data extracted by processing the individual images which are acquired by the camera. In this way, the propagation of errors in local measurements can be bounded. Moreover, CML enables the construction of a photo-mosaic while the vehicle simultaneously localizes itself on this photo-mosaic map, and it is done by using only relative observations.

Apart from the methodology to reduce the accumulation of incremental misalignments, photo-mosaicking systems can be further classified depending on the technique they use to detect motion between the video images. The two main approaches could be divided in feature-based and featureless methods. The first detect features in one image, and these features are then matched in subsequent images (*e.g.* (Gracias, 1998; Trucco, 2000a; Garcia, 2001a; Marks, 1995)). However, matching these features is a difficult task in underwater images. The latter approach is based on minimizing the sum of the squared intensity errors over all corresponding pairs of pixels (Eustice, 2000) or applying differential techniques to estimate motion at each pixel of the image, producing a dense flow field. An evolution of flow methods can be found in the *direct method* for motion estimation (Negahdaripour, 1998j, 1998e, 1999), which allows the estimation of 3D motion directly from spatio-temporal image gradient, without the need of any intermediate measure. However, the differential nature of flow methods requires a high frame rate and negligible changes between consecutive frames. This inconvenience is normally undertaken by introducing a multi-resolution pyramidal scheme. Among featureless methods, a third approach could be considered, based on the detection of motion in the frequential domain (Rzhanov, 2000). Although frequential techniques have been less present in the literature, it may lead to satisfactory results as shown by Rzhanov *et al.* in (Rzhanov, 2000).

In addition, the use of photo-mosaicking systems has to face many other problems. These systems can only be used when the vehicle is performing tasks near the ocean floor and require a reasonable visibility in the working area. Moreover, most photo-mosaicking systems take for granted that the underwater terrain is planar. This hypothesis can be relaxed in the sense that the differences in depth of the seabed are assumed to be negligible with respect to the average distance from the camera to the seabed. Work is being carried out to find adequate solutions to cope with this limitation (Garcia, 2001b; Howland, 2000).

In this paper we propose a feature-based approach to construct photo-mosaics of the ocean floor which follows the methodology of *Concurrent Mapping and Localization*. We upgrade our earlier work (Garcia, 2001a; Garcia, 2000; Garcia, 2001d) by improving the motion estimation methodology. The photo-mosaic image is taken into account as the vehicle evolves by selecting a reference image extracted from the photo-mosaic. Our technique to select the reference image could be considered a mixture of: 1) the extraction of an image, warped from the mosaic, proposed by Negahdaripour *et al.* in (Negahdaripour, 1999), and 2) the fixed-distance interval presented by Marks *et al.* in (Marks, 1995) to add a new image to the mosaic. Moreover, we revisit the use of textural parameters to solve the correspondence problem in underwater images (Garcia, 2001c).

This paper is organized as follows. First, an overview of the proposed methodology for constructing visual mosaics is outlined. Then, section 3 describes the core of our system: the *mosaic engine*. The *engine* is controlled by the *mosaic supervisor*, which is described in section 4. In section 5, the experimental results with real underwater sequences are presented. In order to evaluate the accuracy of the photo-mosaicking system, some experiments based on a laboratory setup are also reported. Some concluding remarks are summarized in section 6, and the last section presents the future directions of our work.

2 Overview

Our proposal of a photo-mosaicking methodology is divided into two main blocks, namely: *mosaic supervisor* and *mosaic engine*. The *mosaic supervisor* keeps the state of the mosaicking system and makes decisions according to this state. It takes into account the readings of on-board sensors, analyses how the vehicle is moving and generates the pertinent orders for the *mosaic engine*. The interface between both blocks is shown in Figure 1. The *supervisor* module is responsible of the mosaic data structure, that is, updating the photo-mosaic image (I_m), and providing the *engine* with the two images which have to be registered. One of these images is the current frame acquired by the camera (I_c). The second one is a reference image (I_r) which has been extracted from the photo-mosaic. Along with these images the engine provides an initial estimate of the apparent motion between them. This initial estimate of the apparent motion is materialized in the form of the "a priori" estimation matrix ${}^{r}\mathbf{H}'_{c}$. This matrix can be computed from the data provided by the on-board sensors (*e.g.* altimeter sonar, compass, Doppler Velocity Log, Inertial Navigation System, inclinometer, etc.). If on-board sensor information is not available, the *supervisor* generates ${}^{r}\mathbf{H}'_{c}$ based on the vision system. The output of the mosaic *engine* is another matrix, ${}^{r}\mathbf{H}_{c}$, which provides a refinement of the initial motion estimation and its associated uncertainty. The *engine* is controlled by the mosaic *supervisor*. Therefore, it is only executed when the *supervisor* requires an iteration of the *engine*, providing a matrix ${}^{r}\mathbf{H}_{c}$ which describes the planar motion between images I_c and I_r.

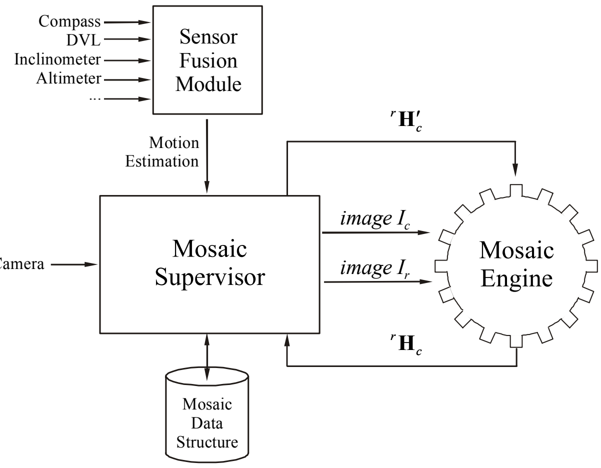


Fig. 1. Dataflow exchanged by the *mosaic supervisor* and the *mosaic engine*.

It should be noted that the description of how the sensor fusion module generates the estimation matrix and its associated covariance beyond the scope of this paper. We will just mention that this sensor fusion operation is performed within the frame of a Kalman filter, and the characterization and identification of the behavior of every sensor is a topic on its own. At the moment, we have only used the measures provided by an altimeter sonar and a compass so far, although we plan to add a DVL in the near future. The present paper is focused on which decisions are taken by the *mosaic supervisor* upon the information provided by the “a priori” matrix, and how the *mosaic engine* performs image processing to construct a photo-mosaic, while the vehicle simultaneously localizes itself in the photo-mosaic.

Our approach can be classified as a feature-based mosaicking system, since the *engine* uses point features to solve the correspondence problem between images I_c and I_r. The operations performed by the *supervisor* can be further divided, as illustrated in Figure 2. First, image I_c is acquired by the camera, and lighting inhomogeneities are compensated through local equalization. Then, a correction of the geometric distortion caused by the lens and the interface of the camera housing is performed. The next step consists of selecting image I_r to be passed jointly with image I_c to the *engine*. Finally, matrix $^{r}\mathbf{H}'_{c}$ is computed and the *mosaic engine* is told to execute. The *engine* then begins its execution by detecting interest points in image I_c, and matches their correspondences in image I_r. Next, the system identifies the points which describe the dominant motion of the image by means of a robust outlier-detection algorithm. Once the pairs of features describing the dominant motion have been selected, a 2D projective transformation matrix relating the coordinates of both images is computed. When the *engine* completes its execution, it provides matrix ${}^{r}\mathbf{H}_{c}$ and the control is given back to the mosaic *supervisor*. The mosaic *supervisor* then decides whether image I_c should be merged with the composite photo-mosaic image or, on the contrary, should only be taken into account to update the positioning estimation of the vehicle. Finally, absolute 3D position estimates are obtained by fusing the information provided by an altimeter sonar and that of the photo-mosaic.

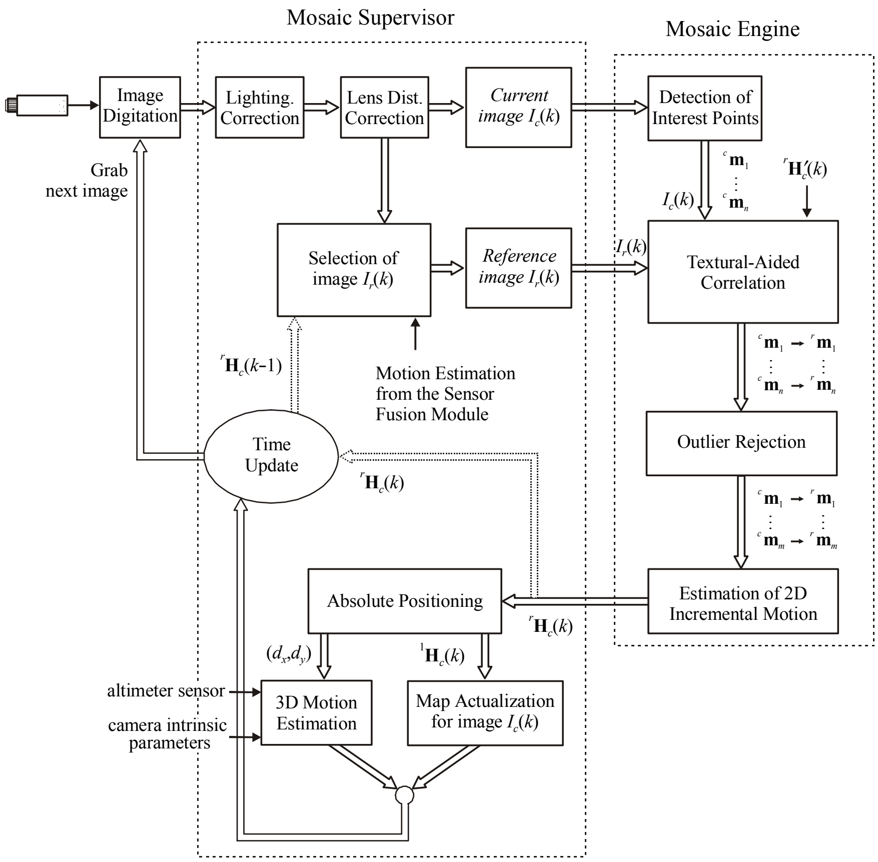


Fig. 2. Block diagram of the proposed mosaicking system.

3 Mosaic Engine

3.1 Selection of Interest Points

The goal of our interest point detector is to find stable features in the image, *i.e.* scene features which can be reliably found when the camera moves from one location to another and lighting conditions of the scene change somewhat. The strategy which has been used to detect these interest points is based on the use of a corner detector. This corner detector finds areas of high variation of the image gradient by using first-order image derivative approximations (Harris, 1988; Kitchen, 82). Normally, small windows containing high frequencies are quite adequate since they are located in the border of different image textures. For this reason, our interest point detector searches for small zones presenting high spatial gradient information in more than one direction. To do this, the image is convolved with two high-pass directional filters, obtaining the image gradient in the x and y directions, that is I_x and I_y. Then, in order to find the areas with the highest gradient in both x and y directions, we follow the same strategy proposed by some corner detectors (Harris, 1988) or feature trackers (Shi, 1994), consisting of computing the point-to-point product of images I_x and I_y in all their combinations. The resulting images ($J_x = I_x I_x$, $J_y = I_y I_y$ and $J_{xy} = I_x I_y$) are then smoothed with a 3×3 Gaussian mask. Finally, a "cornerness" measure based on the work of Harris and Stephens (Harris, 1988) is considered, as shown in equation (1).

$$c = \frac{J_x + J_y}{J_x J_y - J_{xy}^2} \tag{1}$$

The "cornerness" is then computed for all the pixels of the image. High values of parameter c imply poor image gradient. On the contrary, as c gets smaller, the "cornerness" of this area augments. Next, for a considered neighborhood, any values which are not the minimum of the neighborhood are suppressed. This leads to a sparse map of interest points for each image. Finally, since c provides a measure of the quality of the corner, a descending sorting is performed. In this way, the algorithm is able to provide the best n interest points in the image.

3.2 Detection of Correspondences

Once the interest points have been detected in image I_c, the next step consists of finding their correspondences in the reference image I_r. Before searching for correspondences, both images are smoothed with a 3×3 Gaussian mask. Given an interest point ${}^c\mathbf{m}$ in image I_c, a $n \times n$ region centered at this point is selected as a correlation window. This window is subsampled so that only every q^{th} pixel of the window is taken into account, reducing the processed pixels to a matrix $m \times m$, where $m = \left((n-1)/q\right)+1$.

Next, a *search window* is defined in image I_r. This window is normally centered on the coordinates provided by the "a priori" motion estimation matrix ${}^r\mathbf{H}'_c$. The product $({}^r\mathbf{H}'_c\,{}^c\mathbf{m})$ gives the position where the search window is centered. Then a correlation score (s) is computed for every point ${}^r\mathbf{m}$ of the search window (Zhang, 1994), located at coordinates (u,v), as shown in equation (2).

$$s\left({}^c\mathbf{m},{}^r\mathbf{m}\right)=\frac{\sum_{i=-a}^{a}\sum_{j=-a}^{a}\left(I_c(x+i\cdot q,y+j\cdot q)-\overline{I_c(x,y)}\right)\cdot\left(I_r(u+i\cdot q,v+j\cdot q)-\overline{I_r(u,v)}\right)}{(2a+1)^2\sqrt{\sigma^2\left(I_c\right)\cdot\sigma^2\left(I_r\right)}},\quad(2)$$

in which $a=(n-1)/2q$, I_c and I_r the current and reference images for which the motion is to be computed, $\sigma^2(I)$ is the variance of the image computed in the correlation window (see equation (3)); and $\overline{I(x,y)}$ is the average of the correlation window in the image as shown in equation (4).

$$\sigma^2(I)=\frac{\sum_{i=-a}^{a}\sum_{j=-a}^{a}I(x+i\cdot q,y+j\cdot q)^2}{\left(2a+1\right)^2}-\overline{I(x,y)}^2, \qquad (3)$$

$$\overline{I(x,y)}=\frac{\sum_{i=-a}^{a}\sum_{j=-a}^{a}I(x+i\cdot q,y+j\cdot q)}{\left(2a+1\right)^2}. \qquad (4)$$

Equation (2) is normalized by substracting the mean and dividing by a factor which takes into account the dispersion of the gray levels in the considered regions. For this reason, this measurement of correlation is very adequate for underwater imaging, where lighting inhomogeneities are frequent. Unfortunately, although equation (2) produces good results in absence of rotations, its reliability decreases as images I_c and I_r present a higher rotational component. On the other hand, it has been proved that the accuracy of subsampling the correlation window is practically the same as using the full window[1] (Giachetti, 2000). This is due to the strong correlation in the gray level of neighboring pixels, producing smooth intensity variations, especially after a low-pass filtering.

Given an interest point ${}^c\mathbf{m}$, its correspondence ${}^r\mathbf{m}$ could be selected as the point which has obtained the highest correlation score. However, experimental work with underwater images has proved that in some cases the right correspondence is not the one with the highest score, although its score is normally among the best (Garcia, 2001c). Therefore, we select the set of matches $\left\{{}^r\mathbf{m}_1, {}^r\mathbf{m}_2, \ldots, {}^r\mathbf{m}_L\right\}$ with a correlation score higher than a given threshold ξ (that is $s({}^c\mathbf{m}, {}^r\mathbf{m}_i) \geq \xi$), and add them to a list of candidate correspondences of ${}^c\mathbf{m}$.

Next, textural analysis is considered to select the correct matching ${}^r\mathbf{m}$ among the L candidate correspondences. Textural analysis consists in characterizing the interest point and the L possible matches by means of a vector ($\mathbf{v}$). This characterization vector encodes information about the local texture of a given interest point. To characterize the interest point, its 11 × 11 neighborhood is subsampled by means of a 3 × 3 window without considering the coordinate where the interest point is located (see Figure 3). Then, texture is computed at every one of the eight selected locations. At every subsampled position, seven texture operators are computed. These texture operators are selected according to their ability to discriminate image features in underwater images. An extensive study has been carried out in order to compare different texture operators (Ila, 05). We have implemented and tested different configurations of some statistical-based texture operators, *e.g.* co-occurrence matrix (Haralick, 1973), energy filters (Laws, 1980), local binary patterns (Ojala, 1996) and contrast features (Ojala, 1999). Seven texture parameters were selected because of their excellent performance in helping to solve the correspondence problem, as well as their reasonable computational times. These operators are summarized in Table 1. For further information relating to why these textural operators were chosen see (Ila, 05).

Therefore, characterization vector $\mathbf{v}$ computes 7 textural values of every selected location around the interest point, giving rise to a vector of 56 textural values. This vector encodes not only the texture of the neighborhood of the interest point, but also information about the spatial distribution of the neighboring textures. It should be noted that interest points are normally located at the boundaries of regions with different textures. For this

[1] Our default parameters are m=21, q=2 and $\xi = 0.75$.

reason, the texture of the interest point itself is not computed since quite probably different textures will be merged in this area.

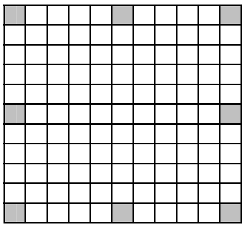


Fig. 3. Subsampling the 11 × 11 neighborhood of a feature. Textural operators are applied uniquely at the gray-level positions (and their corresponding neighborhoods). The computed texture is used to characterize the feature.

Table 1. Textural operators Selected to solve the correspondence problem.

Operator	*Statistical Measure*
L3L3 Energy	3×3 Standard Deviation
E3E3 Energy	3×3 Positive Mean
E3E3 Energy	3×3 Negative Mean
4×4 Co-Occurrence Matrix d=2, θ=45,	3×3 Entropy
L5S5 Energy	3×3 Positive Mean
E5S5 Energy	3×3 Negative Mean
Contrast feature	-

Once a given interest point (${}^{c}\mathbf{m}$) has been characterized by means of the corresponding vector ($\mathbf{v}_c$), its L possible correspondences are also characterized in the same way. The point-to-point Euclidean distance is used as a similarity measure to compare the textural characterization vector of an interest point ${}^{c}\mathbf{m}$ $\left(\mathbf{v}^{(c)} = [v_1, v_2, ..., v_{56}]\right)$ and that of every candidate matching ${}^{r}\mathbf{m}_j$ $(\mathbf{v}_j^{(r)} = [v'_1, v'_2, ..., v'_{56}])$. However, before computing the Euclidean distance, the characterization vectors should be reparametrized, taking into account the minimum and maximum value of the vector in question.

Consider $\mathbf{v}^{(c)} = [v_1, v_2, ..., v_{56}]$ the characterization vector of point ${}^{c}\mathbf{m}$, and $\mathbf{u}^{(c)} = [u_1, u_2, ..., u_{56}]$ the same vector once reparametrized. Then:

$$u_i = \frac{v_i - v_{\min}}{v_{\max} - v_{\min}}, \qquad \forall v_i,\ i = 1..56 \tag{5}$$

where $v_{\min} = \min(\mathbf{v})$ is the smaller value of vector **v**, and $v_{\max} = \max(\mathbf{v})$ is the maximum value of **v**.

The Euclidean distance is then computed between the characterization vector of the interest point and that of every candidate match, as defined in equation (6).

$$d(\mathbf{u}^{(c)}\ ,\mathbf{u}_k^{(r)}) = \sqrt{\sum_{i=1}^{56}\left(u_i - u_i'\right)^2}. \tag{6}$$

It should be noted that the *reparametrization* of a characterization vector does not take into account the distribution of its values. For this reason, it is convenient to weight the distances with the standard deviation σ, as defined in equation (7). The higher the variability of the components of the vector, the higher the deviation and, therefore, distance should be weighted to be less relevant if it has to be compared with other distances. In order to compute this deviation we take into consideration both the elements of the characterization vector of the interest point $\mathbf{u}^{(c)}$ and all the candidate correspondences $\mathbf{u}_j^{(r)}$ found by the region correlation algorithm. Therefore, the weighted distance d_w is computed as follows:

$$d_w = {d}/{\sigma} \tag{7}$$

$$\text{with } \sigma = \sqrt{\sum_{i=1}^{N}\frac{(u_i-\mu)^2}{N} + \sum_{i=1}^{N}\sum_{j=1}^{L}\frac{(u_{ji}'-\mu)^2}{N}} \text{ and } \mu = \frac{\sum_{i=1}^{N} u_i}{N} + \sum_{i=1}^{N}\frac{\sum_{j=1}^{L} u_{ji}'}{N}. \tag{8}$$

Further details of the above methodology can be found in (Ila, 2005). This operation has to be performed for all the candidate matches. Finally, among the L candidate correspondences, the one with the smallest distance d_w is selected as the correct match.

3.3 Outlier Removal and Motion Estimation

After the correspondences have been solved, a set of interest points in the current image I_c and their correspondences in the reference one I_r is obtained. Our aim is now to recover the apparent motion of the camera from these features. This can be done by computing a 2D-transformation matrix

${}^{r}\mathbf{H}_{c}$ which relates the coordinates of a feature in the current frame I_c with its coordinates in I_r, as indicated in equation (9).

$$ {}^{r}\tilde{\mathbf{m}}_{i} = {}^{r}\mathbf{H}_{c} \cdot {}^{c}\tilde{\mathbf{m}}_{i} \text{ or } \begin{bmatrix} {}^{r}x_{i} \\ {}^{r}y_{i} \\ 1 \end{bmatrix} \cong \begin{bmatrix} h_{11} & h_{12} & h_{13} \\ h_{21} & h_{22} & h_{23} \\ h_{31} & h_{32} & h_{33} \end{bmatrix} \cdot \begin{bmatrix} {}^{c}x_{i} \\ {}^{c}y_{i} \\ 1 \end{bmatrix}, \tag{9} $$

where ${}^{c}\tilde{\mathbf{m}}_{i} = ({}^{c}x_{i}, {}^{c}y_{i}, 1)^{T}$ and ${}^{r}\tilde{\mathbf{m}}_{i} = ({}^{r}x_{i}, {}^{r}y_{i}, 1)^{T}$ denote a correspondence point in images I_c and I_r, respectively; the symbol ~ indicates that the points are expressed in homogeneous coordinates; and $\cong$ expresses equality up to scale. The matrix ${}^{r}\mathbf{H}_{c}$ which performs this transformation is known as "homography" (Semple, 1952). This homography assumes a projective model describing the motion between both images. However, matrix ${}^{r}\mathbf{H}_{c}$ could also be defined by a simpler model (Hartley, 2000), decreasing the degrees of freedom (DOF) and, therefore, the number of parameters to estimate (see Table 2). Although experience with real video data acquired by UUVs indicates that better results are normally achieved with the simpler models, the most general model (*projective*) will be used next to illustrate the estimation of the unknowns of ${}^{r}\mathbf{H}_{c}$. Since this matrix is defined up to scale, equation (9) can be written as follows:

$$ \begin{bmatrix} \lambda \cdot {}^{r}x_{i} \\ \lambda \cdot {}^{r}y_{i} \\ \lambda \end{bmatrix} = \begin{bmatrix} h_{11} & h_{12} & h_{13} \\ h_{21} & h_{22} & h_{23} \\ h_{31} & h_{32} & 1 \end{bmatrix} \cdot \begin{bmatrix} {}^{c}x_{i} \\ {}^{c}y_{i} \\ 1 \end{bmatrix} \tag{10} $$

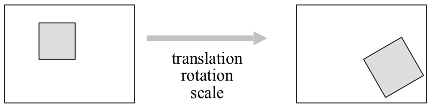
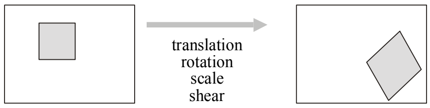
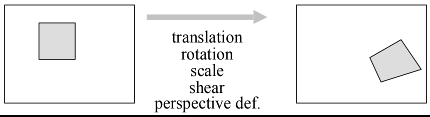
where λ is an arbitrary non-zero constant, and $({}^{r}x_{i}/\lambda, {}^{r}y_{i}/\lambda)$ would be the Cartesian coordinates of ${}^{r}\mathbf{m}_{i}$.

Therefore, solving the homography of equation (10) involves the estimation of 8 unknowns. By using inhomogeneous coordinates instead of the homogeneous coordinates of the points, and operating the terms, the projective transformation of equation (10) can be written as:

$$ \left. \begin{aligned} {}^{r}x_{i} &= h_{11}\,{}^{c}x_{i} + h_{12}\,{}^{c}y_{i} + h_{13} - h_{31}\,{}^{r}x_{i}\,{}^{c}x_{i} - h_{32}\,{}^{r}x_{i}\,{}^{c}y_{i} \\ {}^{r}y_{i} &= h_{21}\,{}^{c}x_{i} + h_{22}\,{}^{c}y_{i} + h_{23} - h_{31}\,{}^{r}y_{i}\,{}^{c}x_{i} - h_{32}\,{}^{r}y_{i}\,{}^{c}y_{i} \end{aligned} \right\} \tag{11} $$

which, expressed in matricial form and considering n pairs point/matching, gives rise to a linear system:

Table 2. Motion models to describe a planar transformation. DOF: degrees of freedom.

Model	Homography Matrix	Distortion
Similarity 4 DOF	$\begin{bmatrix} s\cos\theta & -s\sin\theta & t_x \\ s\sin\theta & s\cos\theta & t_y \\ 0 & 0 & 1 \end{bmatrix}$	
Affine 6 DOF	$\begin{bmatrix} h_{11} & h_{12} & h_{13} \\ h_{21} & h_{22} & h_{23} \\ 0 & 0 & 1 \end{bmatrix}$	
Projective 8 DOF	$\begin{bmatrix} h_{11} & h_{12} & h_{13} \\ h_{21} & h_{22} & h_{23} \\ h_{31} & h_{32} & 1 \end{bmatrix}$	

$$\begin{bmatrix} {}^c x_1 & {}^c y_1 & 1 & 0 & 0 & 0 & -{}^r x_1 \cdot {}^c x_1 & -{}^r x_1 \cdot {}^c y_1 \\ 0 & 0 & 0 & {}^c x_1 & {}^c y_1 & 1 & -{}^r y_1 \cdot {}^c x_1 & -{}^r y_1 \cdot {}^c y_1 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ {}^c x_n & {}^c y_n & 1 & 0 & 0 & 0 & -{}^r x_n \cdot {}^c x_n & -{}^r x_n \cdot {}^c y_n \\ 0 & 0 & 1 & {}^c x_n & {}^c y_n & 1 & -{}^r y_n \cdot {}^c x_n & -{}^r y_n \cdot {}^c y_n \end{bmatrix} \cdot \begin{bmatrix} h_{11} \\ h_{12} \\ h_{13} \\ h_{21} \\ h_{22} \\ h_{23} \\ h_{31} \\ h_{32} \end{bmatrix} = \begin{bmatrix} {}^r x_1 \\ {}^r y_1 \\ \vdots \\ {}^r x_n \\ {}^r y_n \end{bmatrix}. \quad (12)$$

It can be seen that the unknown elements of the homography matrix ${}^r\mathbf{H}_c$ can be computed from equation (12) if 4 or more pairs of matchings are available. The 8 unknowns $\{h_{11}, h_{12}, ..., h_{32}\}$ can be found through *Singular Value Decomposition* (SVD). Given that the accuracy of the estimated parameters also depends on the coordinate frame in which the points are expressed, the normalization strategy proposed on (Hartley, 1997) is carried out. This normalization consists in moving the origin of coordinates to the centroid of the set of points, and scaling the coordinates so that the average distance from any point to the origin is $\sqrt{2}$.

Instead of computing homography ${}^r\mathbf{H}_c$ from all the point/matching pairs, we apply first a robust estimation method, the *Least Median of Squares* (LMedS), to detect the false matches (known as *outliers*) which could still appear among the right correspondences. These "false" correspondences could be in fact correctly matched points which describe a motion which is different from the dominant motion of the image (a moving fish, algae, etc.). The algorithm aims to find a matrix ${}^r\mathbf{H}_c$ which minimizes the median of the squared residuals M_{err} (Rousseeuw, 1987):

$$M_{err} = \underset{j}{med}\left(d^2\left({}^c\tilde{\mathbf{m}}_j, {}^c\mathbf{H}_r\, {}^r\tilde{\mathbf{m}}_j\right) + d^2\left({}^r\tilde{\mathbf{m}}_j, {}^r\mathbf{H}_c\, {}^c\tilde{\mathbf{m}}_j\right)\right), \tag{13}$$

where ${}^c\mathbf{H}_r$ is the inverse of matrix ${}^r\mathbf{H}_c$; and $d^2\left({}^c\tilde{\mathbf{m}}_j, {}^c\mathbf{H}_r\, {}^r\tilde{\mathbf{m}}_j\right)$ is the square Euclidean distance from a point ${}^c\tilde{\mathbf{m}}_j$, defined on current image I_c, to the projection on the same image plane of its correspondence ${}^r\tilde{\mathbf{m}}_j$. Hence, the error residual M_{err} is defined by the distance of a point to the projection of its correspondence and vice versa, the distance from the correspondence to the point, considering the inverse homography ${}^c\mathbf{H}_r$.

Then, the LMedS algorithm consists of computing a candidate solution of matrix ${}^r\mathbf{H}_c$ based on a randomly chosen d-tuple from the data. Then estimate the fit of this solution to all the data, defined as the median of the squared residuals M_{err} of equation (13). The amount of randomly chosen data d may vary depending on the motion model used in the mosaic. For instance, in the case of a general projective homography where 8 parameters have to be estimated, d has a value of 4, while in the case of an affine homography only 6 parameters have to be found, reducing d to 3.

LMedS has a *breakdown point* of 50%, *i.e.*, the algorithm can tolerate up to 50% of outliers, but not more. In principle, all the d-tuples should be evaluated; in practice, a Monte Carlo technique is applied in which only a random sample of size k is considered (Rousseeuw, 1987). Assuming that the whole set of points may contain up to a fraction of ε outliers, the probability that at least one of the k d-tuples consists of d "inliers" (correct data) is given by:

$$P = 1 - \left(1 - (1-\varepsilon)^d\right)^k. \tag{14}$$

Therefore, k can be found from

$$k = \frac{\log(1-P)}{\log\left(1-(1-\varepsilon)^d\right)}. \tag{15}$$

We have selected a probability P = 0.99, and assume a generous percentage of outliers of 40%, which is far beyond the actual number of outliers we will have. Sample size d depends on the motion model used to compute the homography.

As noted in (Rousseeuw, 1987), the LMedS efficiency is poor in the presence of Gaussian noise, considering the *efficiency* of a method as the ratio between the lowest achievable variance for the estimated parameters and the actual variance provided by the given method. To compensate for this deficiency, a weighted least-squares procedure is carried out. The *robust standard deviation* estimate $(\hat{\sigma})$ is obtained through (Rousseeuw, 1987)

$$\hat{\sigma} = 1.4826\left[1 + \frac{5}{n-d}\right]\sqrt{M_{err}}, \tag{16}$$

where M_{err} is the minimal median as defined in equation (13). Once the robust standard deviation $\hat{\sigma}$ is known, a weight can be assigned to each correspondence:

$$w_i = \begin{cases} 1 & \text{if } r_i^2 \le (2.5\hat{\sigma})^2 \\ 0 & \text{otherwise,} \end{cases} \tag{17}$$

in which

$$r_i^2 = d^2\left({}^c\tilde{\mathbf{m}}_i, {}^c\mathbf{H}_r\, {}^r\tilde{\mathbf{m}}_i\right) + d^2\left({}^r\tilde{\mathbf{m}}_i, {}^r\mathbf{H}_c\, {}^c\tilde{\mathbf{m}}_i\right). \tag{18}$$

The pairs of interest point/correspondence having $w_i = 0$ are considered outliers. Therefore, those points at a distance larger than 2.5 times the robust standard deviation are eliminated, and matrix ${}^r\mathbf{H}_c$ is recomputed through SVD with the remaining points. In this way, a new homography matrix ${}^r\mathbf{H}_c$ is robustly estimated.

4 Mosaic Supervisor

4.1 Lens Distortion Removal

Correcting the distortion produced by the camera lenses and the ray diffraction at the interfaces of the water-camera and air-camera housings requires the estimation of a number of intrinsic camera parameters (Xu, 1997). A simplification of the Faugeras-Toscani algorithm has been implemented to correct only radial distortion, instead of performing full camera calibration (Faugeras, 1986):

$$x_u = \left(\frac{x_d - x_0}{k_x}\right) + \left(\frac{x_d - x_0}{k_x}\right)\cdot k_1 \cdot r^2 + c_x \tag{19}$$

$$y_u = \left(\frac{y_d - y_0}{k_y}\right) + \left(\frac{y_d - y_0}{k_y}\right) \cdot k_1 \cdot r^2 + c_y \qquad (20)$$

in which (x_u, y_u) are the ideal undistorted coordinates of the measured distorted point (x_d, y_d), and (c_x, c_y) are the coordinates of the center of the image. The parameters k_x, k_y are the scaling factors in the *x* and *y* directions, respectively. They account for differences on the image axes scaling. The principal point of the image is defined by (x_0, y_0) and represents the coordinates of the projection of the optical center of the camera on the image plane. k_1 is the first term of the radial correction series, and r is the squared distance of (x_d, y_d) from the center of the image and accomplishes:

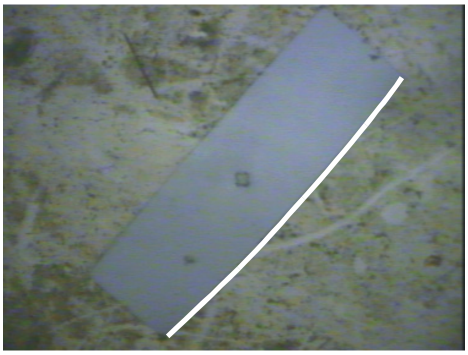
$$r = \sqrt{\left(\frac{x_d - x_0}{k_x}\right)^2 + \left(\frac{y_d - y_0}{k_y}\right)^2} \qquad (21)$$

Once these parameters are known, image correction for radial distortion can be computed (Garcia, 2001d). Figure 4 shows an image taken at our water tank of a metal plate before and after distortion removal.

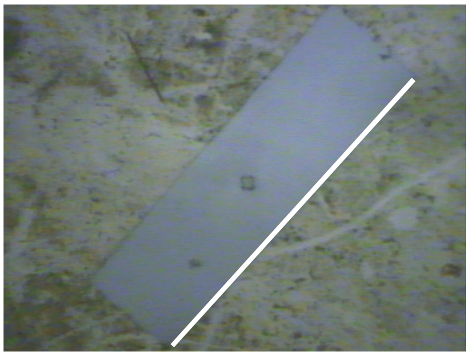
In our implementation, equations (19) and (20) are computed offline. Then, two Look Up Tables (LUT) are stored with the resulting values. At run time, the LUTs are addressed with the undistorted *x* and *y* coordinates and every LUT provides the distorted coordinate from which the gray-level has to be taken.

4.2 Lighting Compensation

Our algorithms have been designed to withstand small changes in the scene illumination. However, compensation of the shading effects produces a better quality of the resulting mosaic. In most cases, the image might be bright in the center and decrease in brightness as one goes toward the edge of the field-of-view. Also, brightness of the scene changes as the vehicle moves. Some authors have proposed the use of local equalization to compensate for the effects of non-uniform lighting (Singh, 1998; Eustica, 2000), darkening the center of the image and lighting the dark zones of the sides. Unfortunately, local equalization is very time consuming and therefore inadequate for real-time applications. Another common technique consists of applying homomorphic filtering using a logarithmic transformation of the images, then high-pass filtering for the resulting image, and finally taking the exponent to restore the image (Gonzalez, 1993). As in the previous case, this set of operations could slow down processing.



(a)



(b)

Fig. 4. Correction of lens distortion in underwater images (a) Image with considerable lens distortion; (b) corrected image. A white line has been overlayed on the contour of the object.

On the other hand, the proposal of Rzhanov *et al.* (Rzhanov, 2000) for removal of lighting inhomogeneities consisted of fitting a two-dimensional polynomial spline to every frame and then subtracting it from the image. This technique considers the lighting of the scene as an additive factor which should be subtracted.

In our approach, we consider the image as a function of the product of the illumination and reflectance properties of a given scene, as described by equation (22).

$$I_{im}(x,y) = I_{ill}(x,y) \cdot I(x,y) \tag{22}$$

where $I_{im}(x,y)$ is the image sensed by the camera, $I(x,y)$ would be the ideal image under absence of shading and $I_{ill}(x,y)$ is the illumination multiplicative factor. We have deliberately ignored the *gain* and *offset* terms which could come from non-uniform camera sensitivity, since in an underwater environment they can be neglected with respect to $I_{ill}(x,y)$. Normally, the lighting field due to light sources carried by the vehicle can be modeled as a smooth function. In order to model this non-uniform illumination, we compute a Gaussian-smoothed version of the captured image $I_{im}(x,y)$. This smoothed image $I_s(x,y)$ is intended to be an estimate of how much the illumination field affects every pixel of the image. To obtain this effect, the smoothing has to be large compared to the size of the objects in the image[2]. Therefore, since illumination acts as a multiplicative factor in the image formation process, an estimate of ideal image $I(x,y)$ can be obtained through

$$\tilde{I}(x,y) = \frac{I_{im}(x,y)}{I_s(x,y)} \cdot \delta \tag{23}$$

where δ is a constant which adjusts the overall image luminance. When we aim to construct mosaics in real time, the smoothed image is not computed for every image of the sequence. It is only computed from the first frames, the result is averaged and then used in equation (23) for every new image, thus saving computational effort.

4.3 Image Selection

The selection of the two images I_c and I_r to be passed to the mosaic *engine* is one of the most important tasks of the *supervisor*. $I_c(k)$ is the current image acquired by the camera (at time instant k), while $I_r(k-1)$ is the previous reference image which was extracted from the mosaic some iterations earlier. From the execution of the algorithm at time $k-1$, the position, scale factor, and orientation of the previous images $I_c(k-1)$ and $I_r(k-1)$ is known with respect to the mosaic reference frame $\{I_m\}$.

A first estimate of the position and orientation of present image $I_c(k)$ with respect to the photo-mosaic $\{I_m\}$ can be obtained from the sensor fusion module. However, if on-board sensing is not available (or it is not enough to provide this information), a rough estimate is derived by having the mosaic *engine* compute the motion between the consecutive images

---

[2] A typical value is 41×41 pixels.

$I_c(k-1)$ and $I_c(k)$. In order to alleviate the computations, this estimate can be computed from a subsampled version of both images. In this way, from the orientation and scaling of the previous image $I_c(k-1)$, a first estimate of the position of $I_c(k)$ in the mosaic reference frame $\{I_m\}$ is obtained (see Figure 5), and is described by matrix ${}^{m}\mathbf{H}'_c$.

$$ {}^{m}\mathbf{H}'_c = \begin{bmatrix} s\cos\alpha & -s\sin\alpha & x_0 \\ s\sin\alpha & s\cos\alpha & y_0 \\ 0 & 0 & 1 \end{bmatrix}, \tag{24}$$

where α is the orientation of image $I_c(k)$ with respect to $\{I_m\}$, (x_0, y_0) represents its displacement, and s corresponds to its scale factor.

Then, if the overlapping of images $I_c(k)$ and $I_r(k-1)$ is below a given threshold (normally 40% of the size of the image), the mosaic *supervisor* will select a new reference image $I_r(k)$. The new reference image will be extracted from the mosaic image $I_m(k-1)$ at the same position and orientation as that of the last image added to the mosaic $I_c(k-1)$. If the overlap between images $I_c(k)$ and $I_r(k-1)$ is bigger than the threshold, the reference image will not change, *i.e.* $I_r(k) = I_r(k-1)$. Once the reference image $I_r(k)$ has been chosen, both $I_r(k)$ and the current image $I_c(k)$ could be passed to the mosaic *engine*. However, as described in the previous section, detection of matchings between images with different orientations is not possible with our mosaic *engine*. To overcome this limitation, image $I_c(k)$ should be rotated so that its orientation be the same as in image $I_r(k)$. In fact, image $I_c(k)$ is not only rotated but also scaled by means of transformation matrix ${}^{C1}\mathbf{H}_c$,

$$ {}^{C1}\mathbf{H}_c = \begin{bmatrix} s\cos\theta & -s\sin\theta & 0 \\ s\sin\theta & s\cos\theta & 0 \\ 0 & 0 & 1 \end{bmatrix}. \tag{25}$$

The orientation angle θ can be computed since the orientation of image $I_r(k)$ was already known, and scale factor s is selected as the scale initially estimated in ${}^{m}\mathbf{H}'_c$. Once transformation ${}^{C1}\mathbf{H}_c$ has been found, we can compute the lacking translation components to create the rotated version $(I_{C2}(k))$ of current image $I_c(k)$.

Image $I_{C2}(k)$ is computed as follows. First, the coordinates of points ${}^{C1}(x_1, y_1)$, ${}^{C1}(x_2, y_2)$ and ${}^{C1}(x_3, y_3)$ illustrated in Figure 5 are estimated by applying ${}^{C1}\mathbf{H}_c$ to the coordinates of the corners of image I_c:

$$\begin{cases} {}^{C1}\begin{pmatrix} x_1 & y_1 & 1 \end{pmatrix}^T = {}^{C1}\mathbf{H}_c \cdot {}^{c}\begin{pmatrix} w-1 & 0 & 1 \end{pmatrix}^T \\ {}^{C1}\begin{pmatrix} x_2 & y_2 & 1 \end{pmatrix}^T = {}^{C1}\mathbf{H}_c \cdot {}^{c}\begin{pmatrix} w-1 & h-1 & 1 \end{pmatrix}^T \\ {}^{C1}\begin{pmatrix} x_3 & y_3 & 1 \end{pmatrix}^T = {}^{C1}\mathbf{H}_c \cdot {}^{c}\begin{pmatrix} 0 & h-1 & 1 \end{pmatrix}^T \end{cases} \tag{26}$$

where $w \times h$ is the size of image $I_c(k)$. Next, the offset terms d_x and d_y, which explain the translation from the coordinate system $\{I_{C1}\}$ to the origin of coordinates of image $I_{C2}(k)$ (see Figure 6(b)).

$$d_x = \begin{cases} min(x_1, x_2, x_3) & \text{if } min(x_1, x_2, x_3) < 0 \\ 0 & \text{otherwise,} \end{cases}$$

$$d_y = \begin{cases} min(y_1, y_2, y_3) & \text{if } min(y_1, y_2, y_3) < 0 \\ 0 & \text{otherwise,} \end{cases} \tag{27}$$

Once offset terms d_x and d_y have been computed, a transformation matrix ${}^{C2}\mathbf{H}_c$ can obtained:

$${}^{C2}\mathbf{H}_c = \begin{bmatrix} s\cos\theta & -s\sin\theta & -d_x \\ s\sin\theta & s\cos\theta & -d_y \\ 0 & 0 & 1 \end{bmatrix}. \tag{28}$$

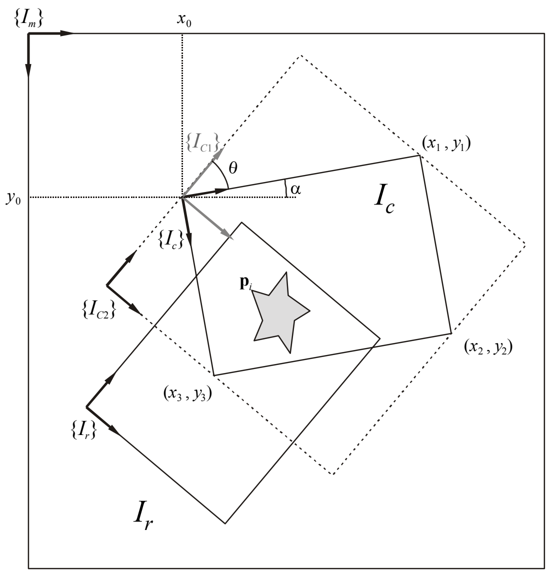


Fig. 5. Warping of present image I_c and a reference image I_r. The present image is rotated so that its orientation matches that of I_r.

Matrix ${}^{C2}\mathbf{H}_c$ is then used to construct image $I_{C2}(k)$ from the current image $I_c(k)$ (see figure 6). In fact, instead of passing image $I_c(k)$ to the mosaic *engine*, it is provided with its rotated version $I_{C2}(k)$. Along with images $I_{C2}(k)$ and $I_r(k)$, an “a priori” estimation ${}^{r}\mathbf{H}'_{C2}$ of the registration parameters between both images must be passed to the *engine*.

The last step consists in finding a matrix which relates a pixel of image $I_{C2}(k)$ with the same point in image $I_r(k)$.

$$ {}^{r}\mathbf{p}_i = {}^{r}\mathbf{H}_m \cdot {}^{m}\mathbf{H}_c \cdot {}^{c}\mathbf{H}_{C2} \cdot {}^{C2}\mathbf{p}_i \text{, with } {}^{c}\mathbf{H}_{C2} = \left({}^{C2}\mathbf{H}_c\right)^{-1}. \qquad (29) $$

Then, matrix ${}^{r}\mathbf{H}'_{C2} = {}^{r}\mathbf{H}_m \cdot {}^{m}\mathbf{H}_c \cdot {}^{c}\mathbf{H}_{C2}$ is the estimation matrix which is passed to the mosaic *engine* module. This methodology has been designed to be used with a strategy of actualization of the photo-mosaic in which every new image only actualizes the mosaic on that areas that have not been updated before, as suggested in (Xu, 1999; Negahdaripour, 1996). In this way, the oldest information (and thus, the one with less accumulated drift) is used to estimate the position of the present image in the mosaic.

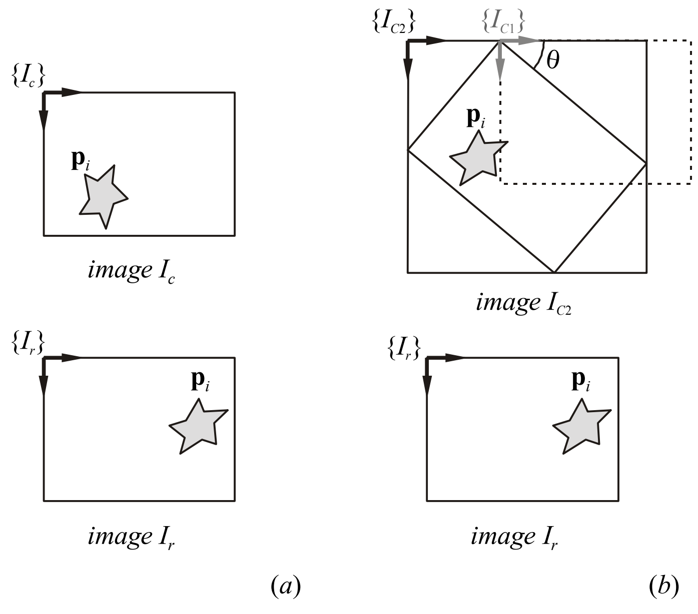


Fig. 6.(a) Original images I_c as captured by the camera and reference image I_r. (b) The present image I_c is rotated and scaled to match the orientation and scaling of I_r, giving rise to image I_{C2}.

When the overlapping is below the threshold, the new reference image incorporates pixels which may have been updated at different time intervals. Image I_r is not extracted from the photo-mosaic at every iteration of the algorithm to give more weight to the visual information coming from earlier images. When a new reference image I_r has to be selected, *i.e.* when there is little overlap between $I_c(k)$ and $I_r(k-1)$, the new reference image $I_r(k)$ is extracted from the mosaic at the orientation of the previously captured image $I_c(k-1)$.

At this point, the control goes to the mosaic *engine*, which has images I_c and I_r, and an "a priori" estimation about the relative motion between both images. In the case where other information coming from the on-board sensors is available, this data can also be used.

4.4 Image Warping and Mosaic Construction

As soon as the best transformation ${}^{r}\mathbf{H}_{c}$ between current $I_c(k)$ and reference $I_r(k)$ frames has been found, the mosaic *supervisor* decides whether image $I_c(k)$ should be warped with the photo-mosaic $I_m(k)$ or, on the contrary, should only be taken into account to update the positioning estimation of the vehicle. To take this decision, the *supervisor* tries to find out how much $I_c(k)$ may contribute to the mosaic. Let $A(k)$ be the overlapping area between images $I_c(k)$ and $I_r(k)$, and let $A(k-i)$ be the overlapping between the last image which was warped to the mosaic $I_c(k-i)$ and its corresponding reference image $I_r(k-i)$. Image $I_c(k)$ will only be added to the photo-mosaic if its superposition $A(k)$ with the reference image is smaller than the overlapping $A(k-i)$ minus a given threshold. This methodology avoids the updating of the mosaic at every iteration of the algorithm when the vehicle moves slowly, and ensures enough updating at normal navigation speed. If image $I_c(k)$ is to be warped with the photo-mosaic, only the new areas which have not been updated before are actualized.

According to our definition of matrix ${}^{r}\mathbf{H}_{c}$, given the homogeneous coordinates of a point ${}^{c}\tilde{\mathbf{m}}_{i}$ in image I_c, the product ${}^{r}\mathbf{H}_{c}\,{}^{c}\tilde{\mathbf{m}}_{i}$ provides the coordinates of this point $\left({}^{r}\tilde{\mathbf{m}}_{i}\right)$ in image I_r. Therefore, it is necessary to find a transformation which maps the reference image with the mosaic frame. Normally, the coordinate system of the first image of the sequence (I_0) is chosen as reference frame (plus a certain offset defined by ${}^{m}\mathbf{H}_{0}$), as shown in Figure 7. Then, the transformation matrix to warp the current image with the mosaic is given by:

$$ {}^{m}\mathbf{H}_{c} = {}^{m}\mathbf{H}_{0}\, {}^{0}\mathbf{H}_{r}\, {}^{r}\mathbf{H}_{c}, \tag{30} $$

where ${}^{m}\mathbf{H}_{0}$ is uniquely the initial translation. Every time a new reference image is extracted from the mosaic, its transformation matrix ${}^{0}\mathbf{H}_{r}$ has to be updated.

4.5 3D Motion Estimation

The aim of this phase is to estimate the position of the vehicle as the mosaic is being constructed, following the Concurrent Mapping and Localization (CML) strategy. At this point, the 2D motion of the camera is known in pixels at every time instant as a *similarity*, *affine* or *projective* measure (rotation, translation, scaling, shear, etc.). With the aid of an ultrasonic altimeter, and the knowledge of the intrinsic parameters of the camera, 3D metric information about vehicle motion can be easily recovered. This is done in the following way. As the distortion produced by the camera lenses and the ray diffractions at the camera housing interfaces has been corrected in the first phase of the mosaicking process, the processed images could be considered an ideal projective projection of the ocean floor. That is, the camera behaves as a perfect *pin-hole* model, producing an ideal linear projection of the incident image rays, as illustrated in Figure 8. Moreover, the vehicle is considered to be stable in pitch and roll due to the

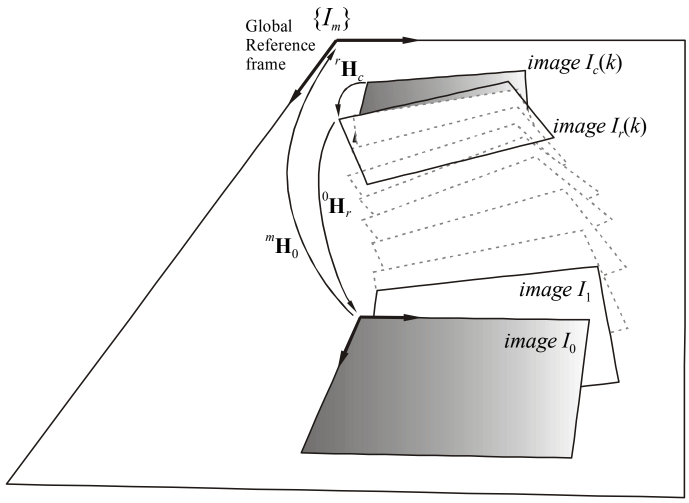


Fig. 7. Mosaic common reference frame. The global registration matrix ${}^{m}\mathrm{H}_{c}$ transforms the image coordinates of any point in the current image $I_c(k)$ to the mosaic image I_m.

distribution of weight (since its center of mass is below its center of buoyancy).

Therefore, the metric measure Z provided by the altimeter, together with the knowledge of the camera focal length f, can be used to convert the motion estimation from the mosaic coordinate system $\{I_m\}$ (in pixels) to the earth reference system $\{E\}$ (metric information). Applying the geometric law of the perspective relation (Kanatani, 1991), the following equation can be obtained:

$$\frac{d}{f} = \frac{D}{Z}, \text{ then } D = \frac{d \cdot Z}{f}. \tag{31}$$

When the first image of the sequence is placed in the mosaic, the earth coordinate system $\{E\}$ is aligned to the XY plane defined by this image, and the initial Z_0 is measured from the altimeter. For every new image, the subsequent homographies provide a 2D estimation of the vehicle motion expressed in $\{I_m\}$. A transformation matrix $^{E}\mathbf{H}_{m}$ can be computed, to convert XY measures in pixels to their corresponding metric coordinates.

$$^{E}\mathbf{H}_m = \begin{bmatrix} \frac{Z_0}{f}\cos\omega & -\frac{Z_0}{f}\sin\omega & 0 \\ \frac{Z_0}{f}\sin\omega & \frac{Z_0}{f}\cos\omega & 0 \\ 0 & 0 & 1 \end{bmatrix}, \tag{32}$$

where ω explains the difference in orientation of the earth coordinate system $\{E\}$ with respect to $\{I_m\}$. Normally, $\omega = \pi$ as shown in Figure 8. The Z coordinate at time instant k is given by the altimeter sonar measure Z_k, while (X_k,, Y_k) are obtained by post-multiplying the coordinates of the central of the current image point in $^{c}\tilde{\mathbf{m}}$ by ($^{E}\mathbf{H}_{m}\,^{m}\mathbf{H}_{c}$).

5 Experimental Results

Two different types of experiments are reported to test our mosaicking system. Firstly, a robot arm carrying a camera ("eye-in-hand" configuration) has been used to scan a poster of the ocean floor. An accurate quantitative measurement of the accumulated error can be obtained by comparing the actual robot trajectory with respect to the trajectory estimated from the photo-mosaic. Secondly, several sea trials have been carried out, under real conditions, using a small UUV with a downward-looking camera.

5.1 Laboratory Setup

The system consists of a robot arm carrying a downward-looking camera (see Figure 9) as well as a poster of the seabed located in the robot workspace. The robot always keeps a configuration where the camera is approximately orthogonal to the poster. The idea of the experiments consists on programming the robot to follow a trajectory and then sense the real trajectory by means of the data provided by the robot controller. Nevertheless, since our robot controller does not provide very accurate information in estimating the position of the end-effector, we decided to estimate the position in the following way. First, we programmed the robot to follow a desired trajectory, replacing the seabed poster by a poster which contains a calibration pattern formed by black dots in a white background. While the end effector describes a trajectory over the calibration poster, a simple image-processing algorithm is used to get the robot position. The algorithm obtains an initial estimation of the executed trajectory by interrogating the robot controller, and this estimate is refined by automatically detecting the black dots of the calibration pattern through the image sequence. Next, the pattern is substituted by a poster of the sea floor and the same trajectory is

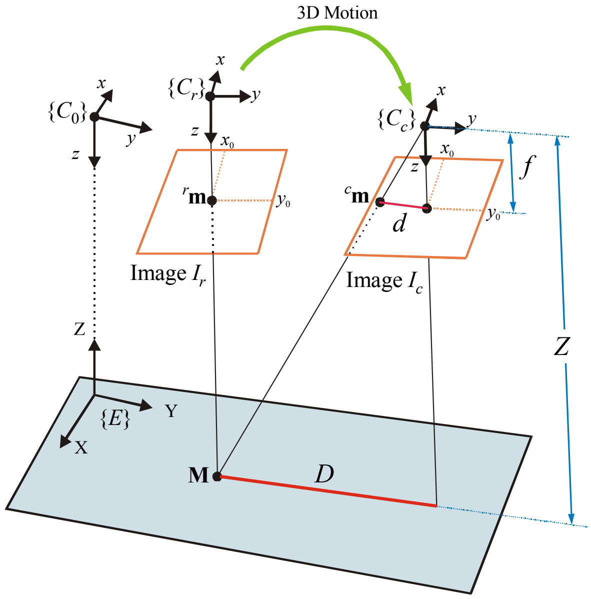


Fig. 8. Motion estimation in Earth (metric) coordinates. The incremental motion d is obtained in pixels from the mosaic. Taking Z from the altimeter sensor, and knowing the camera focal length f, a measure D can be obtained in world coordinates.

executed again. Since our robot arm has very good repeatability, both trajectories can be considered to be the same. Now, the acquired image sequence is used as input to the mosaicking system. Assuming the first trajectory is the real one, the correctness of the mosaic can be quantified.

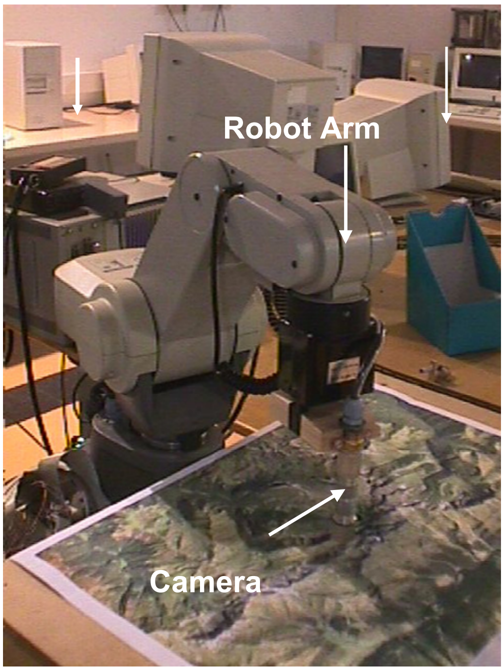
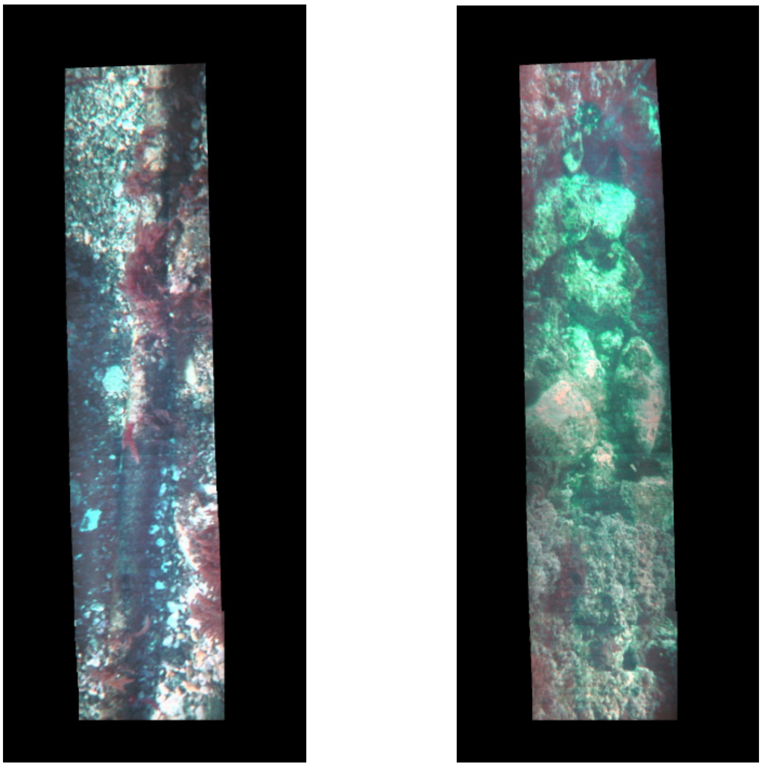


Fig. 9. Experimental setup. A robot arm carries a downward-looking camera and takes images of a poster simulating the sea floor.

5.2 Laboratory Experiments

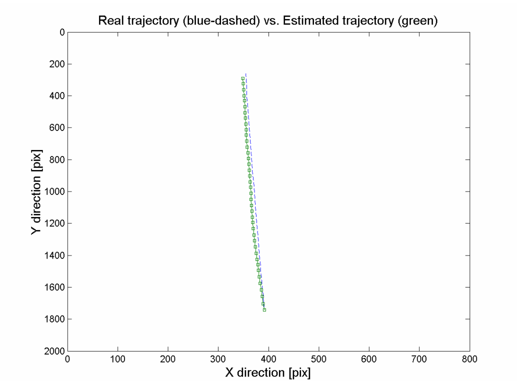
A selected set of three different experiments is presented here, two following a straight path and another one describing an oval trajectory.

In the first experiment, a poster of an underwater pipe surrounded by small stones and some algae has been used. The robot path was chosen to follow the pipe. Figure 10(a) shows the final mosaic and Figures 11(a) and (b) illustrate the real vs. the estimated trajectory as well as the evolution of the drift error. Note that, as expected, the drift increases with time, although it is kept fairly constant from images 11 to 17, with a small correction from frames 18 to 22. This correction is due to the occurrence of errors in the opposite direction.

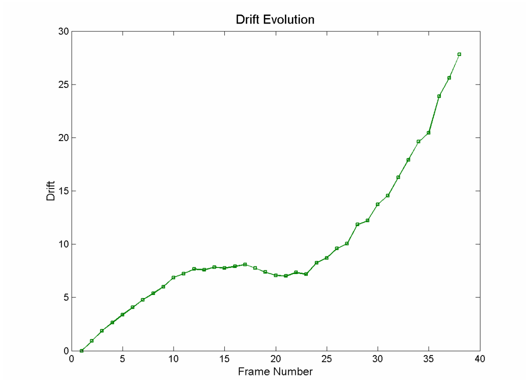


(*a*) (*b*)

Fig. 10. Visual mosaics of a straight trajectory generated from the experimental setup. Both mosaics were created from a sequence of 39 images.



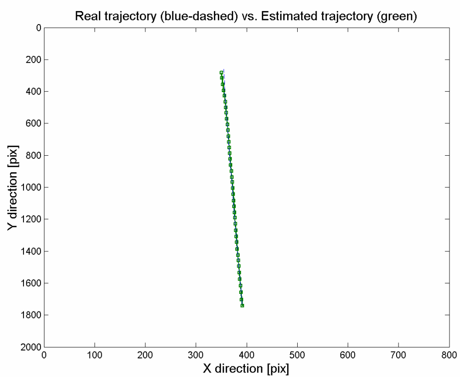
(a)



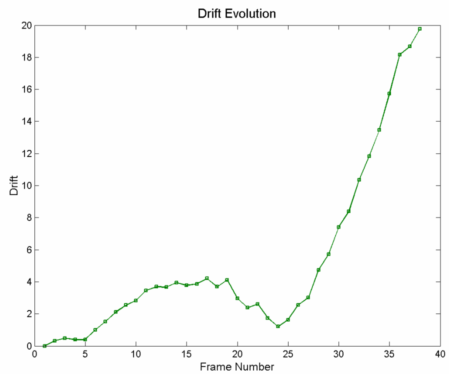
(b)

Fig. 11. Reconstruction of the straight trajectory computed from the mosaic in Figure 10(*a*). (*a*) Real trajectory (dashed-blue) versus estimated trajectory (green with markers). (*b*) drift evolution. Final drift is 27 pixels (1.9% of the total trajectory).

In the second experiment a poster of a typical underwater environment with big rocks was used. This image was taken at more depth, and light turns to the green frequency, decreasing the dynamic range of the image. The Figure 10(b) shows the computed mosaic, while Figures 12(a) and (b) show the real vs. the estimated trajectory as well as the evolution of the drift error. The trajectory estimated through the photo-mosaic follows the real one accurately, although it shortens the trajectory in the last part of the mosaic from frame 25 up to the end of the path. This demonstrates that although some photo-mosaics may have good visual appearance, the estimated trajectory may not be so correct.



(a)



(b)

Fig. 12. Reconstruction of the straight trajectory computed from the mosaic in Figure 10(*b*). (*a*) Real trajectory (dashed-blue) versus estimated trajectory (green with markers). (*b*) Drift evolution. Final drift is 19 pixels (1.3% of the total trajectory).

Finally, in the third experiment, the robot was programmed to follow an oval trajectory on the same poster used for the previous experiment. Figure 13 shows the mosaic and Figures 14(a) and (b) show the real vs. estimated trajectory and the evolution of the drift error. It can be observed that the maximum drift occurs at the end of the path, being lower than a 1% of the total trajectory. We see that from images 76 to 92 the accumulative error produce a reduction of drift. However, from image 93 up to the end of the path, drift increases quite seriously. This proves that in the long term the measurements are uncorrelated, but there is a strong correlation in certain areas, as in the last part of this oval trajectory.

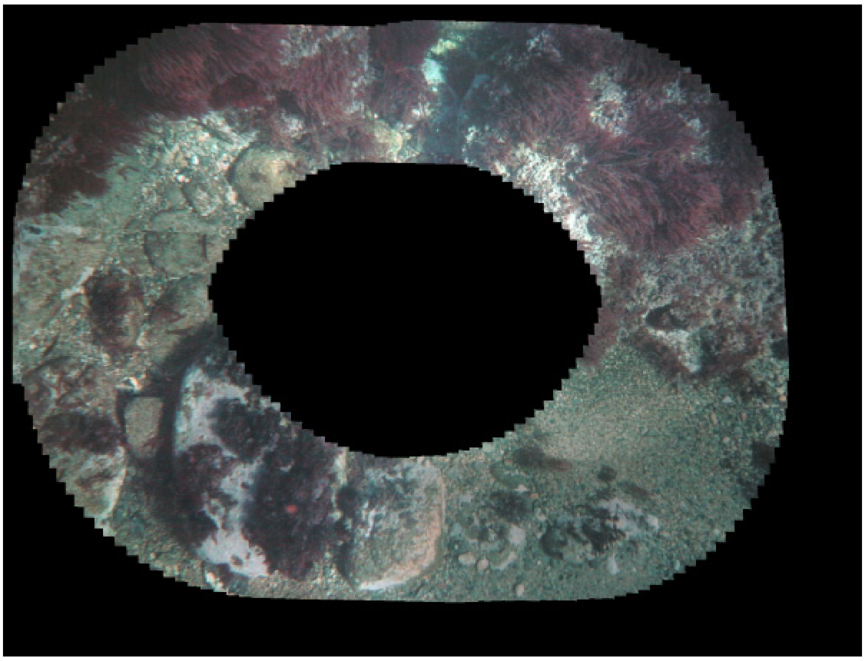
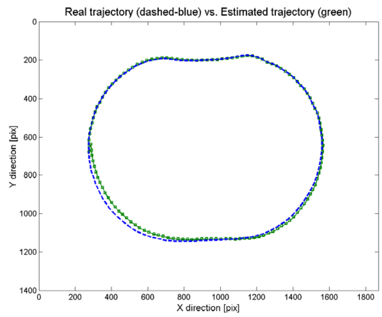
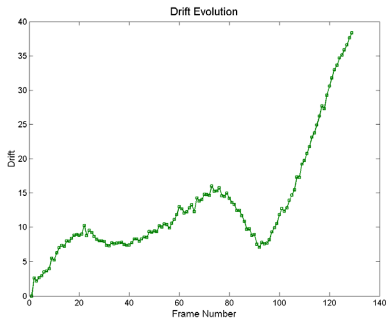


Fig. 13. Mosaic created from a sequence of 130 images. It starts and ends at the center left of the image. A small misalignment can be observed on the left.



(a)



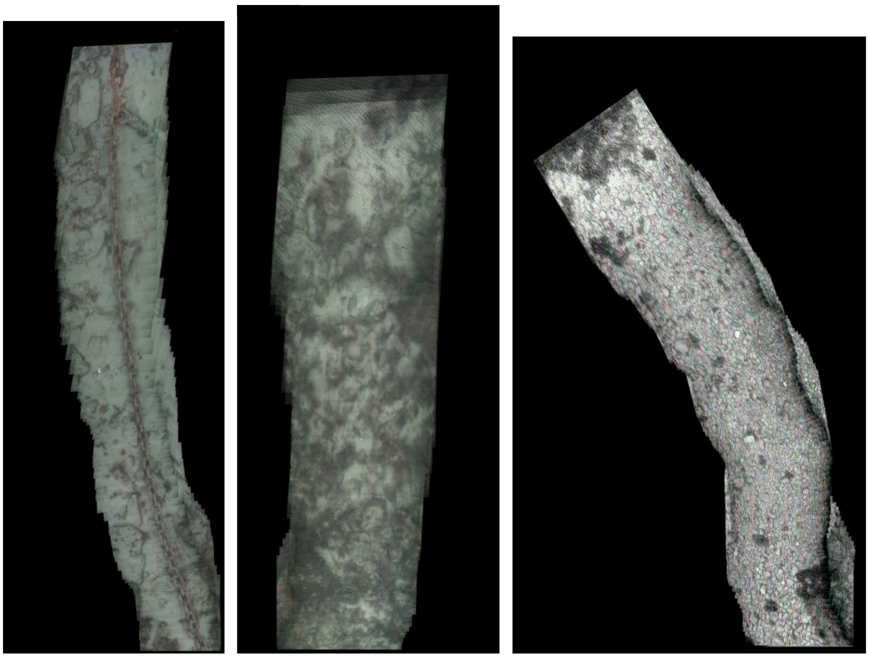
(b)

Fig. 14. (a) Reconstructed trajectory from the photo-mosaic in Figure 13. Estimated (solid with markers) and real (dashed) trajectories followed by the robot arm; (*b*) Drift evolution of the trajectory. The final drift of 37 pixels supposes an error of 1% of the total path.

5.3 Sea Trials

The field tests reported in this paper have been performed in the coastal waters of Costa Brava, using the Unmanned Underwater Vehicle URIS, developed at the *Computer Vision and Robotics Group* of the University of Girona. URIS is shown in Figure 15. To perform each experimental run, the pilot positions the vehicle through teleoperation at a suitable range above the seabed. Then, as the vehicle moves, the acquired images are sent to the surface through an umbilical tether, where they are either stored on a tape or processed in real time. Unfortunately, it is not possible to quantify the errors which are produced in real sea trials since the real trajectory cannot be recovered from any other sensors available in the UUV. All the trials were carried out during the months June-July 2001.

Fig. 15. URIS underwater vehicle.



(a) (b) (c)

Fig. 16. Sample trajectories described by URIS at Costa Brava. These trajectories are from two different sea trials. (*a*) and (*b*) Color mosaics at the entrance of the harbor of Platja d'Aro; (*c*) Trajectory followed at the near the seacoast in Palamós.

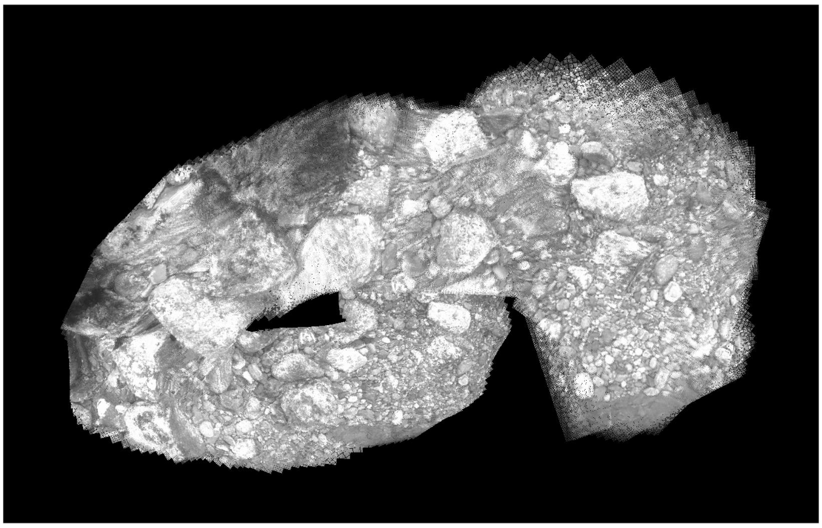


Fig. 17. Simulation of station-keeping in teleoperated mode. Note how the vehicle is able to estimate its position although the underwater relief presents strong variations in depth.

In the first experiment the robot was teleoperated to follow a submerged chain (Figure 16(a)). At the beginning of the sequence the chain is at the range of the seabed, but, as the image sequence progresses, it moves up from the floor. It is possible to see how the underlying assumption of flat scene is violated. However, the vehicle path can be reconstructed from the mosaic without a major problem. Although point correspondences detected in the chain are correctly established, they describe an apparent motion which is different from the background seafloor points due to differences in range. For this reason the robust estimation algorithm detects them as *outliers* and the final motion estimation is visually correct. The affine motion model has been selected in this experiment.

The second experiment shows a trajectory performed in a flat area of the sea floor which presents a slight descent (Figure 16(b)). The vehicle navigates keeping a constant depth, thus augmenting its distance to the seafloor as the sequence progresses. The resulting photo-mosaic has been constructed by considering a projective model. It can be observed how the alignment of the images in the mosaic is quite good. However, although the projective model works fine in flat scenes, it has serious limitations in the presence of microbathymetry, deforming the photo-mosaic in an attempt to accurately align the images. The images at the top part of the mosaic are warped to cover a bigger area of the sea floor, due to the difference in altitude. Since these images are added to the mosaic without interpolation, a few black pixels can be observed in the top of the photo-mosaic.

Figure 16(c) shows another trajectory in a flat area. The mosaic has been constructed using a similarity transformation. The path followed by the vehicle is clearly identifiable from the photo-mosaic. Finally, a last photo-mosaic is shown in Figure 17. In this case the vehicle starts its trajectory at the middle of the photo-mosaic. It then describes a small circle to the left and, as it navigates to the right, it augments its altitude. The surveyed area presents considerable differences in depth (enormous with respect to the navigation altitude). Note that similarity transformations provide good results in non-planar scenes, although homography-based motion estimation methods assume that the observed scene is planar.

6 Concluding Remarks

In this paper we have proposed a new methodology to construct photo-mosaics of the ocean floor. The computational cost of the method allows its use in real missions for the navigation of underwater vehicles. Moreover, a robust solution has been given for the correspondence problem by exploiting the textural characteristics of the selected features. An accurate use of texture information improves, to a large extent, the quality of the detected features.

One of the principal difficulties underwater vision systems have to face is related to the lighting effects. Frequently, the vehicle has to carry its own light source, producing non-uniform illumination, shadows and scattering effects. A technique has been proposed to compensate for these lighting inhomogeneities.

The construction of visual mosaics of the ocean floor can provide accurate position estimates for local navigation of underwater vehicles. However, accumulative error propagates through the image chain as the photo-mosaic increases its size. We have proposed a methodology based on the generation of an adequate reference image extracted from the photo-mosaic, reducing the propagation of errors through the mosaic. An adaptative selection of this image is performed, depending on the motion of the vehicle. The proposed technique is both valid for survey navigation and station-keeping. When the vehicle keeps its station, reference image I_r is kept as a base image which can be rotated and scaled with respect to the live image I_c coming from the camera. However, when the vehicle is navigating and a large displacement exists between consecutive images of the sequence, our methodology results in updating reference image I_r when its overlapping with the present image is smaller than a given threshold.

Moreover, our technique to construct photo-mosaics has been validated by means of an experimental setup, which quantifies error propagation across visual mosaics. This approach is valid for laboratory testing only, but it has also proved helpful in testing and tuning the different parameters of our mosaicking system.

The resulting photo-mosaic is also affected by the selected motion model and the coplanarity of the detected features. When the underwater relief is planar, projective transformations work fine, but in the presence of microbathymetry less general models (like similarity transformations) are able to construct more reliable mosaics.

Fortunately, robust algorithms such as LMedS or RANSAC (Meer, 1991) can, to a large extent, reduce the amount of anomalous data. However, these algorithms have the drawback of random sampling, which is computationally inefficient. Although it does not seriously affect our system since the small sample size (d) leads to a reduced number of samples, the adaptation of other outlier rejection schemes with lower computational costs will be considered in the future.

Most of the mosaicking systems described in the scientific literature compute a planar transformation to register the images. Some other systems estimate 3D motion with respect to a sea floor which is assumed to be planar. However, the mosaicked area may not only be non-planar, but may also present large variations in depth. In this case, new motion estimation and/or 3D structure recovering algorithms should be developed to gain a global perspective of the surveyed zone. This is relevant in the case of the exploration of wrecks or submerged structures with a considerable 3D shape.

Therefore, photo-mosaics can be considered as an important tool which allows motion estimation and local positioning of underwater vehicles at a reduced cost. Although this methodology still has serious limitations (*e.g.* limited survey areas, drift errors, etc), sensor fusion techniques integrating vision with any other sensors will improve in the future the accuracy of the photo-mosaic, and, therefore, improve vehicle navigation.

Acknowledgements

This work has been funded by the Ministerio de Educación y Ciencia projects CTM2004-04205 and DPI2005-09001-C03-01.

References

R. Eustice, H. Singh and J. Howland, “Image registration underwater for fluid flow measurements and mosaicking”, in *Proceedings of the MTS/IEEE OCEANS Conference*, vol. 3, pp. 1529–1534, 2000.

O.D. Faugeras and G. Toscani, “The calibration problem for stereo”, in *Proceedings of the IEEE Computer Vision and Pattern Recognition*, pp. 15–20, 1986.

S.D. Fleischer, H.H. Wang, S.M. Rock and M.J. Lee, “Video Mosaicking Along Arbitrary Vehicle Paths,” in *Proceedings of the OES/IEEE Symp. on Autonomous Underwater Vehicle Technology*, pp. 293–299, Monterey, CA, 1996.

S.D. Fleischer, S.M. Rock and R.L. Burton, “Global Position Determination and Vehicle Path Estimation from a Vision Sensor for Real-Time Video Mosaicking and Navigation,” in *Proceedings of the MTS/IEEE OCEANS Conference*, vol. 1, pp. 641–647, 1997.

R. Garcia, X. Cufí, Ll. Pacheco, “Image Mosaicking for Estimating the Motion of an Underwater Vehicle”, *5th IFAC Conference on Manoeuvring and Control of Marine Crafts*, Aalborg, Denmark, 2000.

R. Garcia, J. Batlle, X. Cufí, and J. Amat, “Positioning an Underwater Vehicle through Image Mosaicking,” in *Proceedings of the IEEE International Conference on Robotics and Automation*, vol. 3, pp. 2779–2784, Seoul, Rep. of Korea, 2001.

R. Garcia, X. Cufi, and M. Carreras, “Estimating the motion of an underwater robot from a monocular image sequence,” in *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems*, vol. 3, pp. 1682–1687, Maui, Hawaii, 2001.

R. Garcia, X. Cufí, and J. Batlle, “Detection of Matchings in a Sequence of Underwater Images through Texture Analysis,” in *Proceedings of the IEEE International Conference on Image Processing*, vol.1, pp. 361–364, Thessaloniki, Greece, 2001.

R. Garcia, J. Batlle and X. Cufi, “A System to Evaluate the Accuracy of a Visual Mosaicking Methodology,” *MTS/IEEE OCEANS Conference*, vol. 4, pp. 2570–2576, Honolulu, Hawaii, 2001

A. Giachetti, "Matching techniques to compute image motion", *Image and Vision Computing*, no. 18, pp. 247–260, 2000.

R.C. Gonzalez and R.E. Woods, “Digital Image Processing”, Addison Wesley, 1993.

N. Gracias and J. Santos-Victor, “Underwater Video Mosaics as Visual Navigation Maps,” *Computer Vision and Image Understanding*, vol. 79, no. 1, pp. 66–91, 2000.

N. Gracias and J. Santos-Victor, “Underwater mosaicking and trajectory reconstruction using global alignment,” in *Proceedings of the MTS/IEEE OCEANS*, vol. 4, 2557–2563, 2001.

N. Gracias and J. Santos-Victor, “Automatic mosaic creation of the ocean floor,” in *Proc. MTS/IEEE OCEANS*, vol. 1, pp. 257–262, 1998.

R. Hartley, and A. Zisserman, “Multiple View Geometry in Computer Vision,” Cambridge University Press, 2000.

R.M. Haralick, K. Shanguman and I. Dinstein, “Textural Features for image classification,” *IEEE Transactions. on Systems, Man and Cybernetics*, vol. 3, pp. 610–621, 1973.

C.G. Harris, and M.J. Stephens, “A combined corner and edge detector,” in *Proceedings of the Fourth Alvey Vision Conference*, Manchester, pp. 147–151, 1988.

R.I. Hartley, “In defense of the eight-point algorithm”, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 19, no. 6, pp. 580–593, 1997.

J.C. Howland and H. Singh, “Simulation of the deep sea mosaicking process”, in *Proceedings of the MTS/IEEE OCEANS Conference*, vol. 2, pp. 1353–1357, 2000.

V. Ila, R. García, X. Cufí, J. Batlle, "Interest Point Characterisation through Textural Analysis for Rejection of Bad Correspondences," *Pattern Recognition Letters.* Vol. 26, No. 10, pp. 1587-1596, 2005.

K. Kanatani, “Computational projective geometry”, *International Journal of Computer Vision*, Graphics and Image Processing, vol. 54, no. 3, pp. 333–348, 1991.

L. Kitchen and A. Rosenfeld, “Gray-Level corner detection,” *Pattern Recognition Letters*, vol. 1, no. 2,pp. 95–102, 1982.

K.I. Laws, “Textured Image Segmentation,” Ph.D. Thesis, Processing Institute, University of Southern California, Los Angeles, 1980.

J.-F. Lots, D.M. Lane, E. Trucco and F. Chaumette, “A 2-D Visual servoing for Underwater Vehicle Station Keeping,” in *Proceedings of the IEEE International Conference on Robotics and Automation*, Seoul, Korea, pp. 2767–2772, 2001.

R. Marks, S. Rock, and M. Lee, “Real-time video mosaicking of the ocean floor”, in *Proceedings of IEEE Symposium on Autonomous Underwater Vehicle Technology*, pp. 21–27, 1994.

R. Marks, S. Rock, and M. Lee, “Real-time video mosaicking of the ocean floor, ” *IEEE Journal of Oceanic Engineering*, vol. 20, no. 3, pp. 229–241, 1995.

P. Meer, D. Mintz, D. Y. Kim, and A. Rosenfeld, “Robust regression methods in computer vision: A review,” *Int. J. Comput. Vision*, no. 6, pp. 59–70, 1991.

S. Negahdaripour, and X. Xu "Mosaic-based positioning and improved motion-estimation methods for automatic navigation of submersible vehicles", *IEEE Journal of Oceanic Engineering*, vol. 27, no. 1, pp. 79–99, 2002.

S. Negahdaripour, X. Xu, and A. Khamene, "Applications of direct 3D motion estimation for underwater machine vision systems", in *Proceedings of the IEEE/MTS OCEANS Conference*, vol. 1, pp. 51–55, 1998.

S. Negahdaripour, S. Zhang, X. Xun and A. Khamene, "On shape and motion recovery from underwater imagery for 3D mapping and motion-based video compression", in *Proceedings of the IEEE/MTS OCEANS Conference*, vol. 1, pp. 277–281, 1998.

S. Negahdaripour, X. Xu, and L. Jin "Direct Estimation of Motion from Sea Floor Images for Automatic Station-Keeping of Submersible Platforms", *IEEE Journal of Oceanic Engineering*, vol. 24, no. 3, pp. 370–382, 1999.

T. Ojala, M. Pietikäinen, and D. Harwood, "A comparative Study of Texture Measures with Classification Based on Feature Distribution," *Pattern Recognition*, vol. 29, pp. 51–59, 1996.

T. Ojala and M. Pietikäinen, "Unsupervised texture segmentation using feature distributions," P*attern Recognition*, vol. 32, pp. 477–486, 1999.

C. Roman, and H. Singh, "Estimation of error in large area underwater photomosaics usgin vehicle navigation data," in *Proceedings of the OCEANS Conference*, pp. 1849–1853, 2001.

P. Rousseeuw and A. Leroy, "Robust Regression and Outlier Detection", John Wiley & Sons, New York, 1987.

Y. Rzhanov, L. Linnett, and R. Forbes "Underwater Video Mosaicing for Seabed Mapping", in *Proceedings of the IEEE Conference on Image Processing*, vol. 1, pp. 224–227, 2000.

J.G. Semple and G.T. Kneebone, "Algebraic projective geometry," Oxford University Press, 1952.

J. Shi and C. Tomasi "Good features to track", in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 593–600, 1994.

H. Singh, J. Howland, D. Yoerger and L. Whitcomb, "Quantitative photomosaicing of underwater imaging," in *Proceedings of the MTS/IEEE OCEANS*, vol. 1, pp. 263–266, 1998.

C.M. Smith, J.J. Leonard, A.A. Bennett and C. Shaw, "Feature-based concurrent mapping and localization for autonomous underwater vehicles", in *Proceedings of the MTS/IEEE Oceans Conference*, vol. 2, pp. 896–901, 1997.

E. Trucco, A. Doull, F. Odone, A. Fusiello and D.M. Lane, "Dynamic Video Mosaics and Augmented Reality for Subsea Inspection and

Monitoring," in *Proceedings of the Oceanology International Conference*, pp. 297–306, 2000.

E. Trucco, Y. R. Petillot, I. Tena Ruiz, K. Plakas and D. M. Lane, "Feature Tracking in Video and Sonar Subsea Sequences with Applications," *Computer Vision and Image Understanding*, vol. 79, no. 1, pp. 92–122, 2000.

L. Whitcomb, D. Yoerger and H. Singh, "Advances in Doppler-based navigation of underwater robotic vehicles," in *Proceedings of the IEEE International Conference on Robotics and Automation*, vol. 1, pp. 399–406, 1999.

X. Xu, and S. Negahdaripour, "Vision-based motion sensing from underwater navigation and mosaicing of ocean floor images," in *Proceedings of the MTS/IEEE OCEANS*, vol.2, pp. 1412–1417, 1997.

X. Xu, and S. Negahdaripour, "Recent advances in vision-based station keeping, navigation and video mosaicking for unmanned submesible vehicles; sea trial experiments", in *Proceedings of the 11th International Symposium of Unmanned Untethered Submersible Technology*, 1999.

Z. Zhang, R Deriche, O. Faugeras, and Q.T. Luong, "A robust technique for matching two uncalibrated images through the recovery of the unknown epipolar geometry," INRIA RR-2273, 1994.

CHAPTER 10

Impact system for underwater ship's hull cleaning

T. AKINFIEV[1], M. ARMADA[1], J. VIBA[2] and R. FERNANDEZ[1]

[1]Instituto de Automatica Industrial – CSIC, Madrid, Spain armada@iai.csic.es; [2] Institute of Mechanics, Riga Technical University, Riga, Latvia. {akinfiev,armada,roemi}@iai.csic.es

An underwater robot intended for cleaning a ship from sea adherence is presented in this chapter. Special attention is paid to the design and control of a cleaning head that uses an impact system. Because synthesis and production of real control systems must foresee all kinds of parameter changes, adaptive control algorithms are proposed. In this way, all kinds of parameters change are taken into account for the synthesis of stable vibroimpact system using adaptive control of spring stuffiness.

1 Introduction

There are about 150 Ship Repairing Yards spread over 27 EU and Associated Countries. Classifying Societies Surveys obliges to inspect ship hull steel two times every five years, with a maximum delay of three years between two inspections. This is mandatory for certifying ship's conditions. On the other hand, it is well known that ship underwater hull, including flat bottom, vertical side and boot-top become overgrown with sea adherence (weed, barnacles) rapidly. The surface covered by sea adherence is about 80% of total ship hull and all sea adherences have to be cleaned before inspection. Besides preventing direct surveying, sea adherence causes a decrease of ship's speed and consequently augments consumption of fuel. For recuperation of ship's performance, it is necessary to dry-dock a ship and to proceed a clean off sea adherence. This cleaning is always required before any other repairing/maintenance activities can follow on. Nowadays cleaning is done manually in dry-dock with an employment of different adapted methods like grit blasting or water jet. It has to be noticed that, in itself, it is a very contaminant operation (the resulting dust always contains painting particles), it is harmful for human operators health and it is a very uncomfortable job.

In contrast to above mentioned, the use of the new technical solution, described below, allows cleaning a ship from sea adherence without the use of an expensive dry dock. In this case it would be possible to clean a ship much more often, for example, twice a year and, consequently, a ship could be in a much better state at all times. Even in the case when it is necessary to make inspection, repair or restoration of a protective coat of a ship in a dry dock, it would be possible to make a ship's hull cleaning out of a dry dock first, which would allow using dry dock more efficiently. Such approach will allow to increase capacity of dry dock at about 15% (average time of ship's staying in dry dock is seven days, of which one day is usually spent for sea adherence cleaning).

The chapter is organised as follows. After the description of the underwater robot configuration, the main part of the chapter concerns cleaning head design and control. Synthesis and production of real control systems must foresee all kinds of parameters changes: change of machine mechanical parameters, change of processing media parameters, and change of machine control system parameters. For this reason systems with adaptive control may be used (Viba, 1998)(Panovko et al., 1997)(Panovko et al., 1998) (Akinfiev et al., 2002). It is shown how all kinds of parameters change will be taken into account for the synthesis of stable vibroimpact system using adaptive control of spring stuffiness.

2 Configuration of Underwater Robot

Underwater climbing robot AURORA is a transportation device for carrying on board cleaning instrument and sensors for steel surveying (Akinfiev et al., 2001). One of the most important problems during the underwater cleaning process is the environment protection. Special system for disposal of rubbish and its utilisation is needed. This device must include flexible guard of working zone of cleaning, a system of removal of waste from working zone with the help of intensive flow of water (water vacuum cleaner), and a system of separation of waste from water with following waste recovery.

The underwater robot is made as a platform 1 with three wheels 2 (see Fig. 1). Each wheel 2 has a power drive 3. Each motor is supplied with angle-of-turn sensor (encoder), connected with a control system. Front wheel is fixed with a possibility of a turn relative to platform 1 and has a steering drive 4 for a realization of a turn.

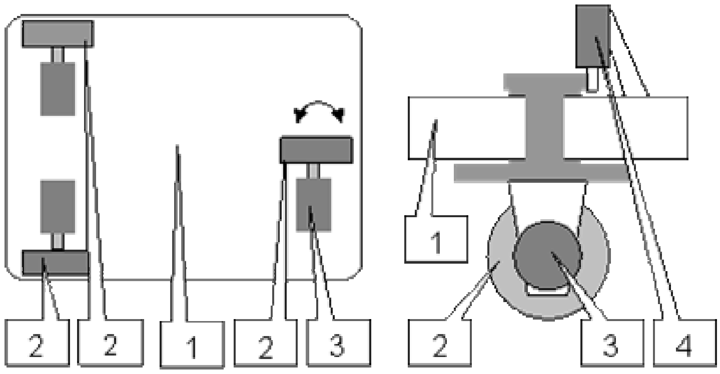


Fig. 1. AURORA robot platform.

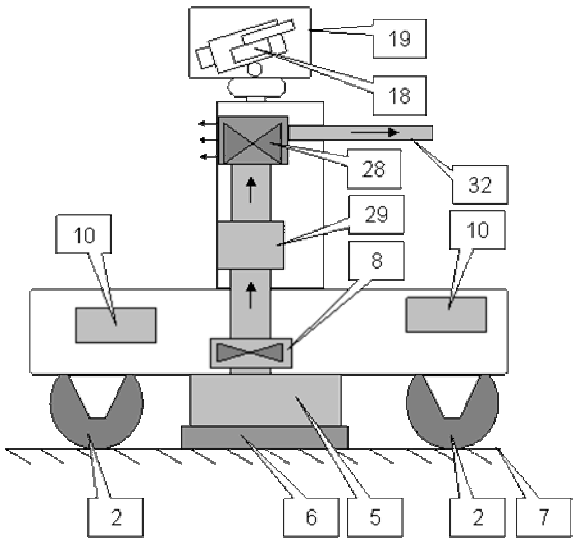


Fig. 2. Configuration of AURORA underwater robot

The box 5 is fixed on a platform of the robot. This box is opened at the bottom, supplied with a flexible sealant 6 (see Fig. 2). The sealant 6 is in contact with a surface 7, on which the robot moves. The box 5 is connected with an extraction pump 8 through a hose 32 (for water pumping off and maintenance of low pressure inside the box 5).

Buoyant elements 10 are fixed on the robot. For the robot more reliable keeping of on a surface, the size and location of the buoyant elements are selected in such a manner that the gravitational force of the robot is completely compensated by buoyancy force affected the robot (when robot is completely immersed in the water). In other words, the magnitude of a gravitational force of the robot is equal to magnitude of buoyancy force, and the points of application of a gravitational force of the robot and the point of application of buoyancy force coincide.

A control system and a control panel (Akinfiev et al., 2001) are located outside of water (on a coast, or on that ship, which surface is cleaned by the robot, or on a special boat).

3 Cleaning Head

Inside the box 5 there is a device for cleaning (see Fig. 3) with several brushes, supplied with a drive. Brushes are destined to clean soft sea adherences (Kyosuke, 1983). For hard sea adherences, the device of cleaning (see Fig. 3) contains a scraper 13 of a material (for example, wood), which hardness is less than the hardness of a surface cover. The base 15 of the scraper is fixed on the body 1 of the robot. The scraper is made of several independent parts. Each part is pressed against the working surface 7 with the help of the corresponding spring element 14. The scraper is supplied with a drive 16, which realizes vibrating displacements of the scraper relative to its base. Some other methods are considered of hard sea adherences cleaning.

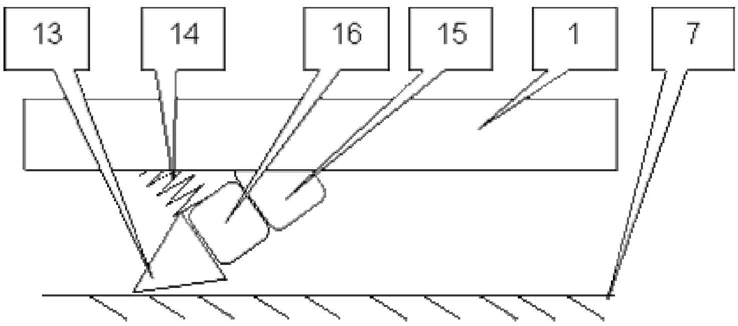


Fig. 3. Device for cleaning hard sea adherences

4 Magnet Gripper

To increase the force that retains the robot on a surface and to provide a possibility of robot's movement out of the water (partial or complete), the robot contains a magnetic element fixed on the body of the robot with a possibility of magnet element interaction with a surface, on which the robot moves (Hirose et al., 1992). The magnetic element can be made as a few independent magnets. At fixing magnetic devices immediately on a body of the robot, a distance between magnets and a working surface can vary considerably because the surface has a curvature with a radius that can change essentially. This leads to the fact that grasping force would vary considerably. To avoid this negative effect, all wheels are made coupled, the parts of the wheel are fixed rigidly on one axis with a clearance, and inside this clearance there is magnet (see Fig. 4). As a tricycle scheme of the robot is employed, each of wheels is always pressed against the working surface, even if there is a curvature. In this case, a distance between magnet and working surface remains practically constant, which ensures a constant value of grasping force. Let us notice that during operation of the robot there arise additional pushing forces due to rotation of brushes (Bernoulli effect) and to a pressure decrease inside a cleaning head at the expense of the water and waste exhaust from this zone.

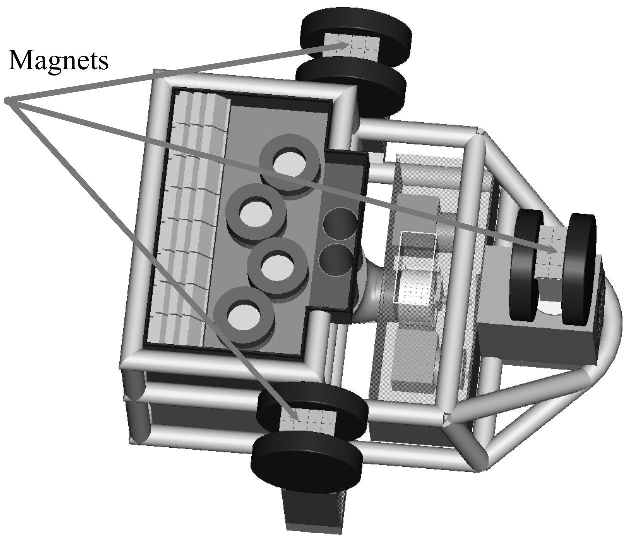


Fig. 4. Magnets location

5 Protection of Environment

When the robot for cleaning of a surface is working, it is necessary to collect the agglomerated depositions. For this purpose robot has a special device. This device is moving off the water with waste from the zone inside the closed box. Then the waste is transported into a special container, located outside the robot.

For ensuring of wastes transporting it is necessary to first crushing them. For this purpose a deposition grinder is used. The deposition grinder 29 (see Fig. 2) is fixed on the robot, supplied with a drive, for depositions deleted from the surface during a cleaning process, and this grinder is inserted into a breakage of a flexible tube that binds the pumps.

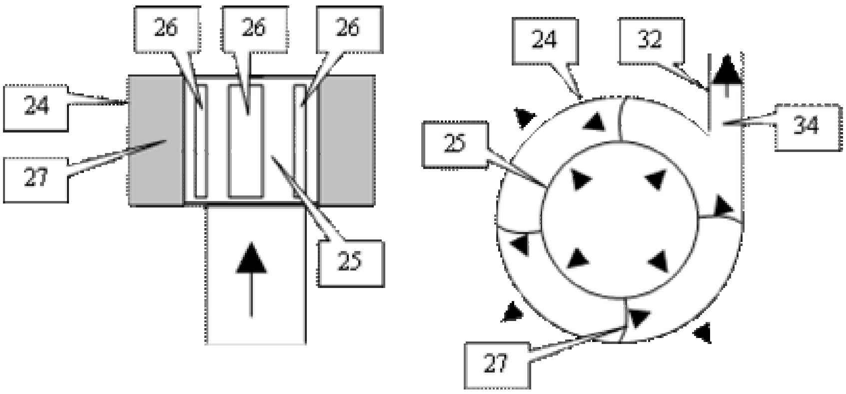


Fig. 5. Configuration of filter-separator

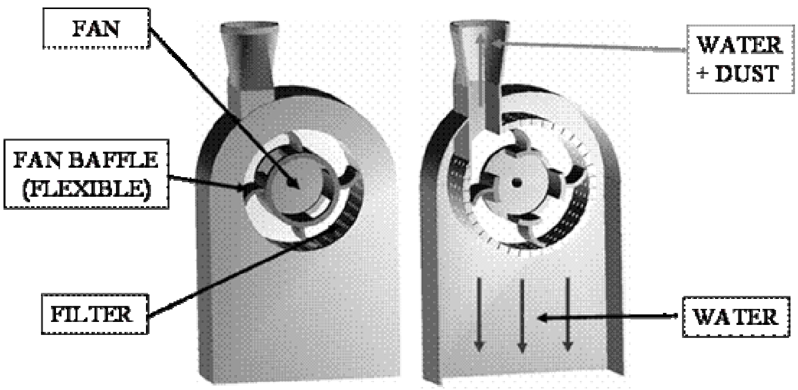


Fig. 6. Filter-separator

To create a water flow; which is capable to evacuate wastes, two pumps are used, which are sequentially installed and connected among themselves by flexible tube. One pump 8 (see Fig. 2) is situated on the robot, and another - outside of the robot (on a coast or on a boat).

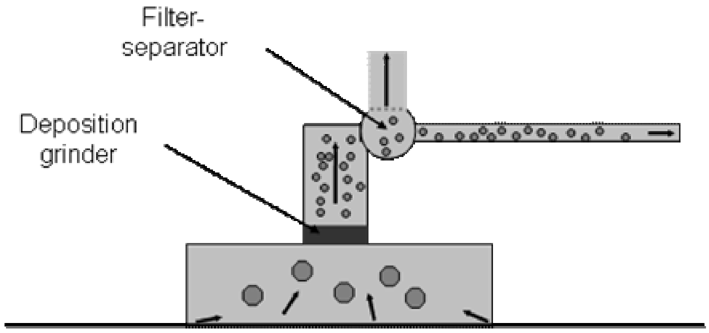


Fig. 7. Moving away of sea adherences

For reduction of energy consumption while transporting the wastes, the pump, fixed on the robot, has to have much greater water flow rate, than the other pump, situated outside the robot. A special filter-separator 28 (see Fig. 2) is fixed on the robot supplied by a cleaning device. This filter is installed in a breakage of a flexible tube 32 that connects the pumps, and the external surface of a gauze element is opened (see Fig. 5, 6) (directly into the ambient medium or through a special hose). The filter element 24 is made cylindrical, and the device of cleaning of the filter is a drum 25 (with holes 26) supplied by a motor. This drum rotates around the axis that coincides with the axis of the cylindrical filter, and has flexible blades 27, fixed on the drum. These blades can contact with a filter surface. The exit hole of the filter is opened, which allows ejecting the wastes with a small portion of water through a flexible tube 32. As a matter of fact, such filter-separator works as a partial flow filter, as a filter permitting to eject the excessive water, and as a centrifugal pump forcing in a flexible tube the wastes with a small portion of water (see Fig. 7).

6 Model of Cleaning Head

Cleaning system has many components of motion (Akinfiev et al., 2003) (see Fig. 8). Part of them is provided by slowly motion of machine body. Cleaning head (subsystem) has one vibroimpact motion across processing media (see Fig. 9). This vibroimpact motion is investigated here with aim to ensure stable processing if the main random parameters are: - length of working element (diminish or fracture); - thickness of processing media (gap); - hardness and dissipation of processing media; - fluctuation and delay of control system switch points; - uncertain resistance forces of system.

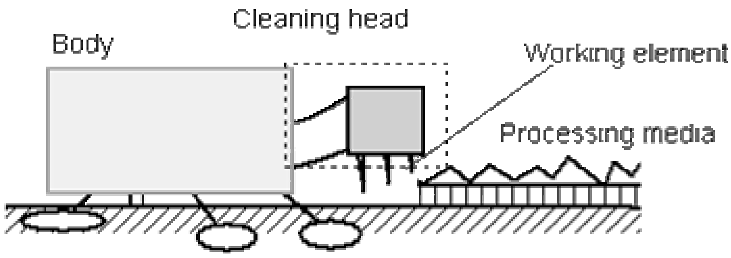


Fig. 8. Scheme of system.

7 Equation of Motion for Simple Model

For system with one degree of freedom equation of motion without random parameters is given by:

$$m\ddot{x} + (C1 + C3 + \Delta(x,\dot{x})) \cdot x = P0 + F(x,\dot{x}) + (1 - \frac{x-\delta}{|x-\delta|}) \cdot 0{,}5 \cdot R(x, \qquad (1)$$

where m - mass; x - displacement; $\dot{x}$ - velocity; $\ddot{x}$ - acceleration; $C1, C3$ – stiffness of springs; $P0$ - constant force given by the springs initial tensions or cleaning head weight ; $F(x,\dot{x})$ - resistance forces; δ - gap when $x = 0$; $R(x,\dot{x})$ - processing force (see Fig. 9).

For adaptive control (excitation) is used change of common spring system stiffness $\Delta(x,\dot{x})$ in the way (see Fig. 10) (Viba, 1988):

$$\Delta(x,\dot{x}) = \frac{C2}{2}(1 - \frac{x \cdot \dot{x}}{|x \cdot \dot{x}|}) \cdot \qquad (2)$$

where C2 is a constant.

Theoretical motion and control action without random parameters is shown in Fig. 10. Respectively the results of motion modelling from the rest position in this case are shown in Fig. 11 and 12. For modelling was used: - nonlinear (cubic and dry friction) resistance forces; - nonlinear (cubic) processing media resistance forces; - linear and nonlinear (cubic) processing media elastic forces.

Analysis of motion character allows concluding that trajectories in the phase plane do not cross and transient process is very short. That is appropriate quality for parametrical resonance systems with adaptive control (Viba, 1988).

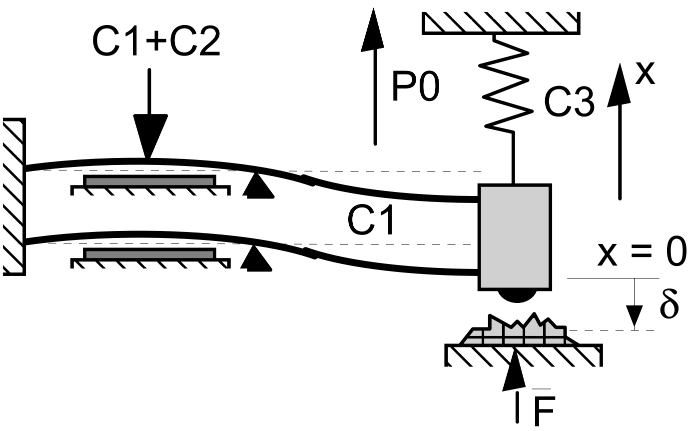


Fig. 9. Scheme of subsystem

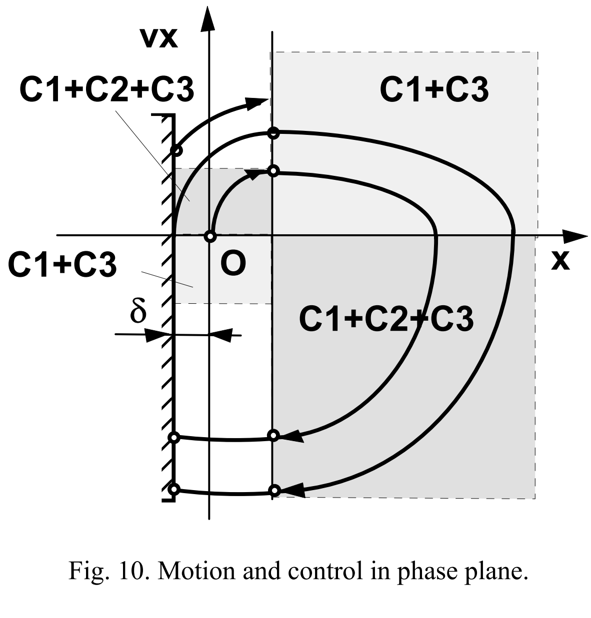


Fig. 10. Motion and control in phase plane.

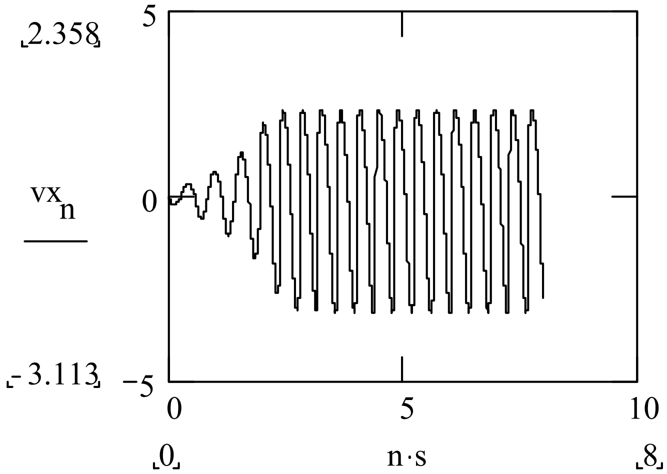


Fig. 11. Velocity in time domain without random parameters.

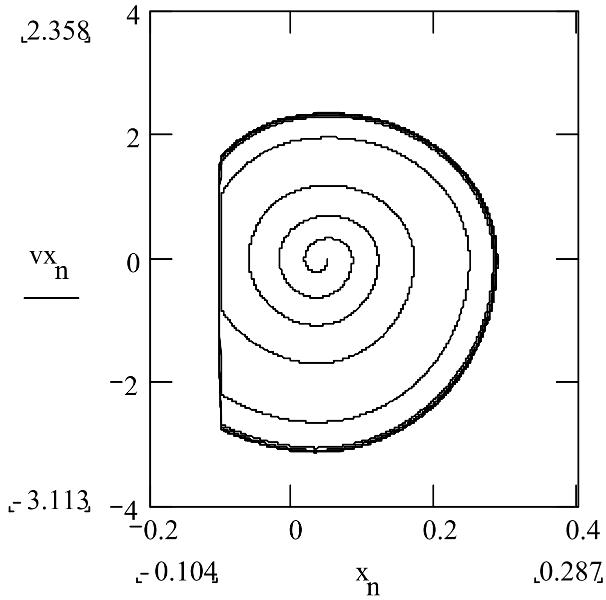


Fig. 12. Motion in phase plane without random parameters.

8 Modelling with Random Parameters

Equation of motion is the same (1) with difference, that some parameters are changing in time or random. When diminish of the working element take place the system is very stable too (Fig. 13., 14). Similarly nothing happen when the working element fracture take place (only change levels of phase coordinates x and $\dot{x}$) (Fig. 15, 16).

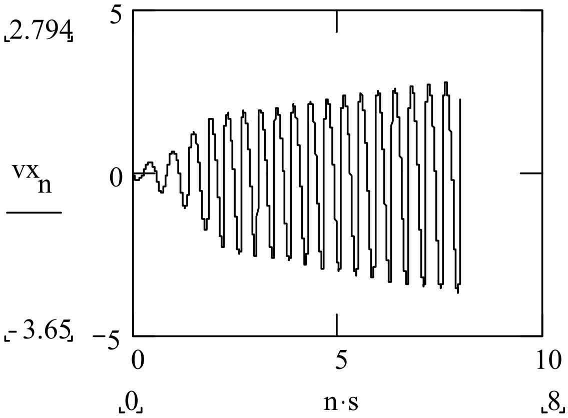


Fig. 13. Velocity in time domain with diminishes of working element.

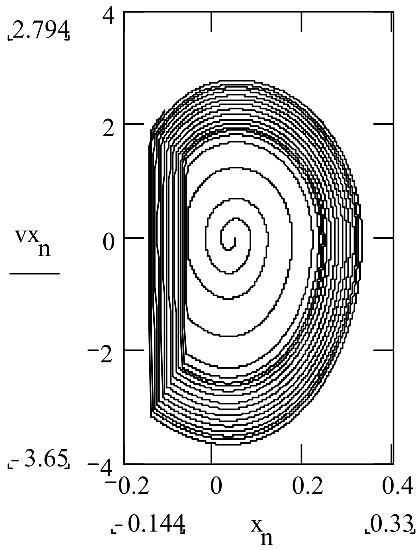


Fig. 14. Motions in phase plane with diminish of working element.

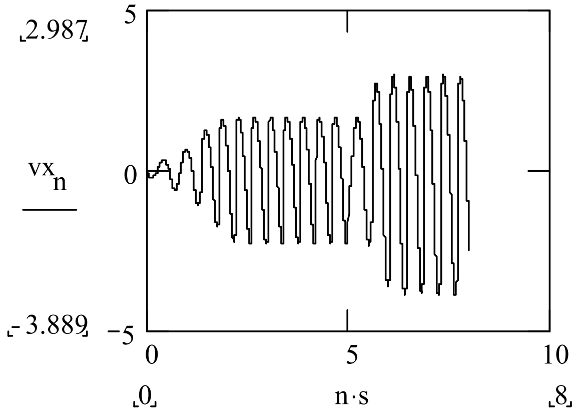


Fig. 15. Velocity in time domain with fracture of working element.

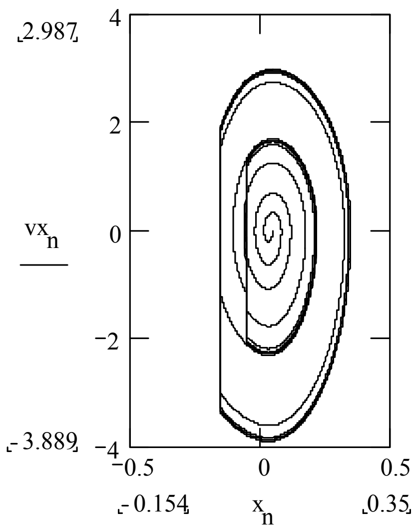


Fig. 16. Motion in phase plane with fracture of working element.

The same very well results about motion stabilization where obtained for change of gap and random hardness and dissipation of processing media (see Fig. 17 - 20).

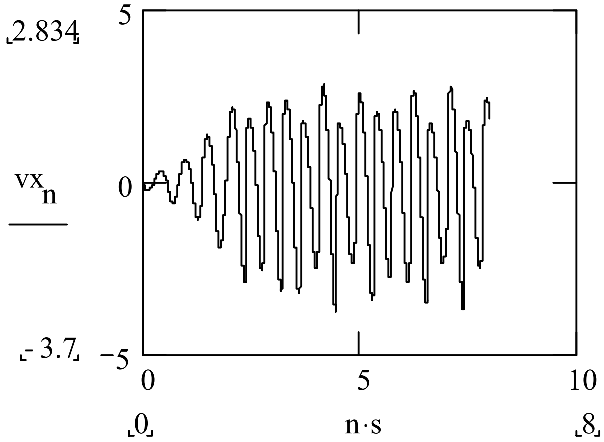


Fig. 17. Velocity in time domain with random gap.

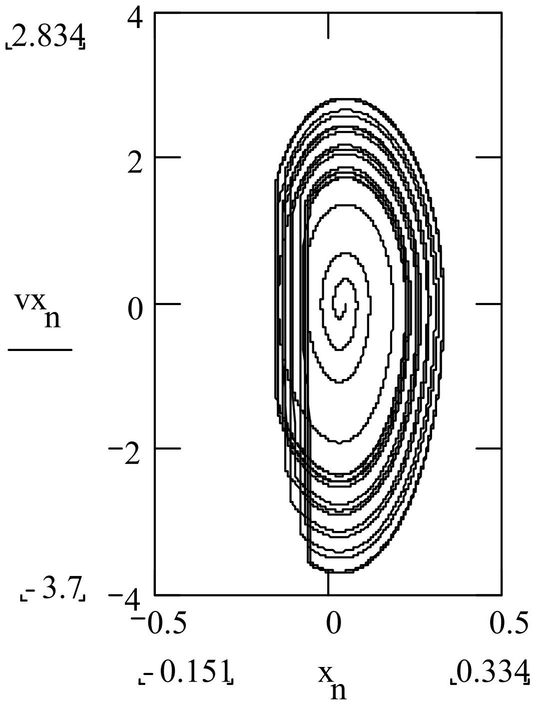


Fig.18. Motion in phase plane with random gap.

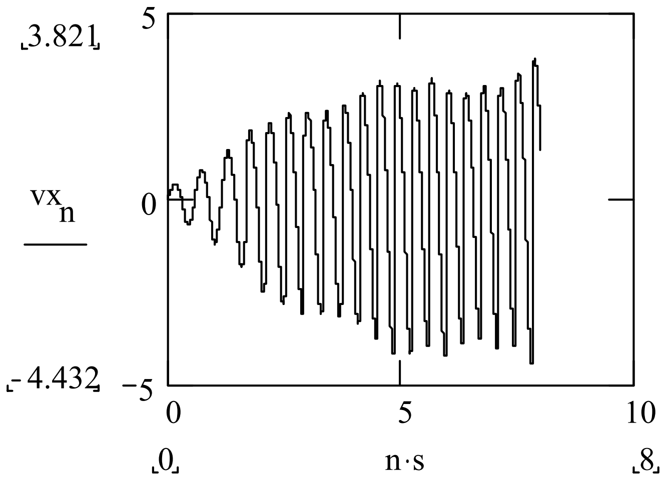


Fig. 19. Velocity in time domain with random hardness and dissipation of processing media.

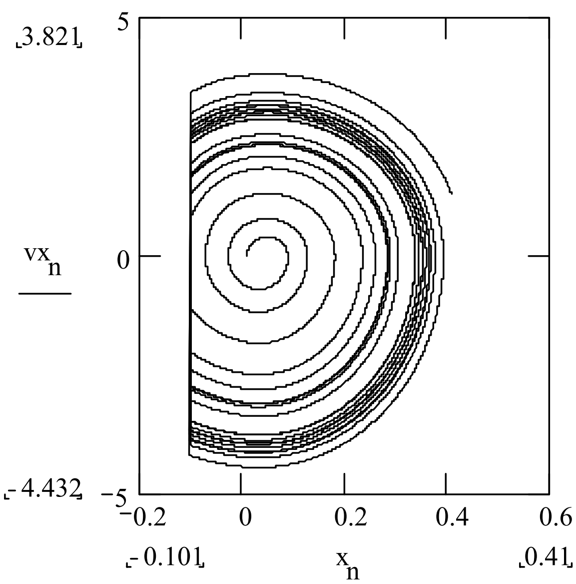


Fig. 20. Motion in phase plane with random hardness and dissipation of processing media.

The fluctuation in time of control action was investigated in a case when system has zero gap and switch command delay occurs when velocity is positive $\dot{x}>0$ (see Fig. 21). Analysis shows that then impact velocity decrease and may be motion without collisions against cleaning surface (Fig. 22). Investigations show too that if the gap is negative is problem with motion start from cleaning surface. Therefore for real systems is recommended to choose positive gap.

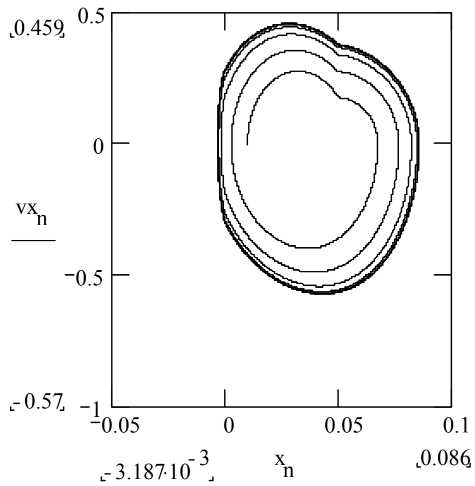


Fig. 21. Delay of switch command.

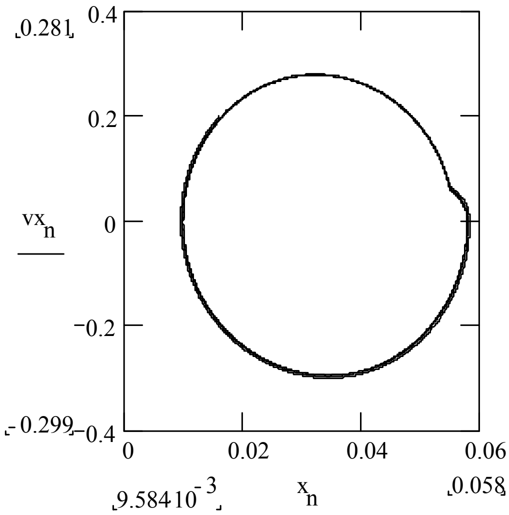


Fig. 22. Motion without collisions.

9 Conclusions

In this chapter a concept of a special robot was presented, allowing making underwater ship hull cleaning operations in a safer and innovative way. The use of such robots will enable exclude the use of dry dock for performance of this operation, which will reduce many times the cost of ship cleaning. Furthermore, a simple and other more complex model with random parameters and adaptive control for an impact cleaning head were studied. The models of simulation take into account all changes of systems and excitation parameters. Investigation shows that systems control with parametric excitation is advisable because trajectories in the phase plane do not cross and transient process is very short. That results in appropriate quality for parametrical resonance systems with adaptive control. The study is a part of project for the development of new cutting and cleaning systems.

Acknowledgements

The authors would like to acknowledge the support of the European Commission, who funded this research in the frame of the Competitive and Sustainable Growth Programme Project GRD1-1999-11153 "Auxiliary Climbing Robot for Underwater Ship Hull Cleaning of Sea Adherence and Surveying-AURORA". This research was also partially funded by CICYT under TIC2000-2576-CE. Acknowledgement is extended to AURORA Partners: two ship-repairing yards, T. Kalogeridis&Co. Inc. and Union Naval de Barcelona, Algosystems S.A., the Division of Robotics from Lund University, SAIND, manufacturer and vendor of equipment for shipyards, and Riga Technical University.

The authors would also like to acknowledge the financial support from Ministry of Science and Technology of Spain (“Ramón y Cajal” Project “Theory of optimal dual drives for automation and robotics”), and from Ministry of Culture and Education of Spain (FPU Fellowship). Also authors would like to extend the acknowledgement to the CICYT (Ministry of Education) for the funding of AUTOMAR Network and to Consejería de Educación (Comunidad de Madrid) for funding RoboCity2030 (S-0505/dpi-0176) project.

References

Akinfiev T., Armada M., Gonzalez P., Ros S. 2001. Underwater robot and control algorithm. Patent application P200102543.

Akinfiev T., Armada M., Ros S., Fernández R. 2001. Concept of underwater robot with multiprocessor control system. Journal Artificial Intelligence, Nauka (Ukraine), (3): 474-480.

Akinfiev T., Panovko G., Viba J. 2002. Start stop drives with adaptive control. Proceedings of Transport and Engineering. Mechanics (Riga Technical University), (7):71. - 78.

Akinfiev, T., Armada, M., Viba, J. 2003. Synthesis of impact cleaning system with adaptive control. Proceedings XIV International Symposium "The Dynamics of Vibroimpact (strongly nonlinear) system", Moscow-Zvenigorod, Russian Ac. of Sciences, (1):pp. 3-7.

Kyosuke Edahiro. Development of underwater robot cleaner for marine live growth in power station. ICAR'83 International Conference on Advanced Robotics. Tokyo, 1983, pp. 99-106.

Hirose, S., Tsutsumitake, H., Toyama, R., and Kobayashi, K. 1992. Disk Rover: A wall-climbing robot using permanent magnet disks. Proc. IEEE/RSJ Int.Conf. on Intelligent Robots and Systems, 2074-2079.

Fernández R., Akinfiev T., and Armada M. 2002. Control algorithms for an underwater climbing robot. Proceedings of the 5th International Conference on Climbing and Walking Robots, Professional Engineering Publishing, London, 985-992.

Panovko G,. Akinfiev T., Armada M., and Viba J. 1997. Adaptive control for reliable oscillated motion in a system with one degree of freedom. International Scientific and Technical Conference. Reliability and Safety of Technical Systems. October 28 - 29, Minsk, 1997. - p. 14 16

Panovko, G., Berestnev O., Viba J., and Grasmanis B. 1998. Setting up of Resonance in Self Controlled Oscillator. Solid Mechanics and its Application. Proceedings of IUTAM / IFToMM Symposium, Riga, Latvia, (73):219-226.

Viba, J. 1988. Optimisation and Synthesis of vibro-impact machines. Zinatne, Riga,(in Russian).

CHAPTER 11

Sea trials simulation

E. LÓPEZ[1], F. J. VELASCO[2], E. MOYANO[3] and T. M. RUEDA[2]

[1]Departamento de Ciencias y Técnicas de la Navegación, Máquinas y Construcciones Navales, Universidad del País Vasco (UPV/EHU). eloy.lopez@ehu.es
[2]Departamento de Tecnología Electrónica, Ing. de Sistemas y Automática, Universidad de Cantabria. velascof@unican.es
[3]Departamento de Matemática Aplicada y CC. de la Computación, Universidad de Cantabria. emiliano.moyano@unican.es

This chapter describes the simulation of the behaviour of two ships in some of the most widely used sea trials: turning circle, pull-out, zig-zag, and direct and inverse spiral test. For the simulation, a full non-linear mathematical model with three or four degrees of freedom is used. The hydrodynamics coefficients used in the mathematical models were obtained from a physical scale models with a planar motion mechanism in a towing tank. The results obtained are satisfactory, supporting the proposal that these sea trial simulation tools should be used as an important part of the design stage in the building of a ship.

1 Introduction

Sea vessels must be able to maintain their course in the open sea, to manoeuvre safely in ports and restricted channels and to stop within a reasonable distance. These minimum capacities are required under any load condition, both at high speeds and at more moderate speeds associated with restricted waters and in both calm conditions and in windy or rough conditions.

This chapter describes the sea trials widely used to determine a ship's manoeuvring characteristics. With these tests, it is possible to measure the ship's dynamic behaviour characteristics, to obtain an indication of its straight-line stability, to evaluate the robustness and the limitations of the control system and to assess the ship's behaviour in emergency situations.

Although these tests were carried out on a ship actually built and at sea, the ship's behaviour can be simulated in the design stage by means of simulation programmes using mathematical models. This paper presents the results of the sea trials of two ships. The simulation is performed with Matlab's Simulink simulation programme.

2 Full-scale Maneuvering Trials

Many of the sea maneuvering trials performed on most merchant ships before they are formally delivered to the ship owner are based on the verification of the maximum speed of the ship, on the functioning of the steering and radio communications systems, on the main engine and the ship steering equipment. In order to verify the ship's maneuvering characteristics, other standard ship manoeuvres can be performed allowing the ship's dynamic behaviour characteristics to be measured and the robustness and the limitations of the ship control system to be evaluated. The manoeuvring characteristics can be obtained by holding or changing a predetermined course and speed in a systematic way.

In accordance with the recommendations of the 14th "International Towing Tank Conference" (14th ITTC, 1975) and other resolutions of the International Maritime Organization (IMO) (Haro, 2004), tests have to provide owners and builders with information on the operating characteristics of the ship. These must address the course-keeping, course changing and emergency manoeuvre characteristics. In order to determine the efficiency of the vessel in course-keeping, the tests methods proposed are: the direct or reverse spiral test and the zigzag manoeuvre test with small rudder angle. To determine the quality in the course changing behaviour, the zigzag manoeuvre test and the 15 degrees helm turning test and change of heading test are recommended. To determine the ship's capacity in the face of emergency manoeuvres, the most appropriate test methods proposed are the maximum helm turning test and the crash-stop astern test.

Vessels must have manoeuvring capacities which allow them to hold course, turn, test the turns, operate at an acceptably low speed and stop in a satisfactory way. Sea maneuvering trials are intended to provide a measurement of the following characteristics (Lewis, 1989):

- *Turning circle characteristics:* These can be determined by means of turning circle tests using a rudder angle of 35º.

- *Yaw checking ability:* This can be measured by obtaining the first overshoot angle and time to check the yaw in a zig-zag manoeuvre.
- *Initial turning ability:* This can be determined at the initial stage of the zig-zag manoeuvre with the ship's change of heading angle per unit of rudder angle, and the forward distances covered after executing a rudder command.
- *Coursekeeping ability:* No single measurement of coursekeeping ability has yet been developed. However, in the case of vessels whose type, size and speeds are comparable, this capacity can be evaluated through a comparison of the zig-zag direct or reverse spiral and pull-out tests.
- *Slow steaming ability:* The capacity to proceed at a steady slow speed is a desirable characteristic. It is generally determined using only the ship's speed associated with the lowest possible engine RPM.
- *Stopping ability:* This can be calculated using the distance that the ship travels along its track, once the crash-astern command has been given.

A description is given below of some of the standard ship manoeuvres: turning circle, pull-out, zig-zag and direct and reverse spiral tests.

2.1 Turning Circle

This is the manoeuvre which has received the most attention from professionals in the field. It is used to determine the ship's steady turning radius and to verify the behaviour of the steering machine and rudder control during course-changing manoeuvres. Figure 1 shows a turning circle, indicating its characteristic stages and parameters. It should be performed to both port and starboard at maximum speed, with a maximum rudder angle and with a rudder angle of 15 degrees. It is necessary to do a turning circle of 540º at least to determine the main parameters of this trial.

This manoeuvre is also used to determine other characteristic parameters such as: the tactical diameter, advance, transfer, loss of speed on steady turn, and times to change heading 90º and 180º respectively, as can be seen in Figure 1. The maximum advance and maximum transfer can also be measured.

As can be observed in Figures 1 and 2, the turning circle test is devised in the following phases:

- *Approach phase:* The ship sails in advance, in a straight line at a constant speed U and with the rudder in neutral position ($\delta = 0$). The linear and angular velocities and accelerations are: $v = \dot{v} = r = \dot{r} = 0$
- *Manoeuvre phase:* This begins when the constant rudder angle δ is applied at any of the sides. It is divided into three phases:

First phase: Starts at the instant the rudder begins to deflect from the neutral position and finishes when it reaches the desired δ value. During this stage, the speeds are practically null (v ≈ r ≈ 0). However, the accelerations have a value of $\dot{v} < 0$ and $\dot{r} > 0$ from the first moment.

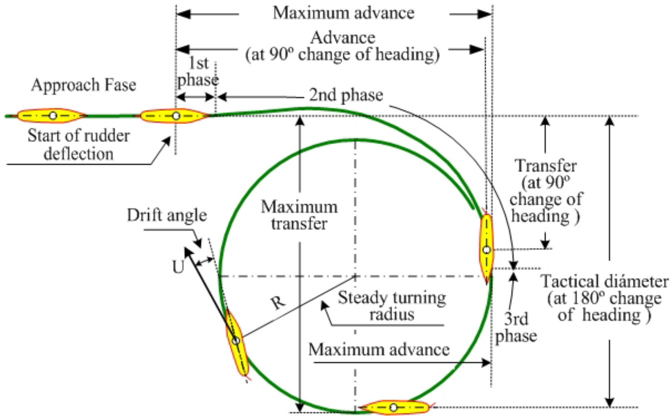


Fig. 1. Terms used in turning circle.

Second phase: Here, the accelerations coexist with the speeds, that is, $v \neq 0;\ r \neq 0;\ \dot{v} \neq 0;\ \dot{r} \neq 0$. In the last part of this stage, equilibrium is obtained between the forces intervening in the ship's turning circle and the accelerations tend to reduce to zero.

Third phase: When this equilibrium is reached, the ship begins to perform a turn of constant radius R, as shown in Figure 1. In this phase, $v \neq 0;\ r \neq 0;\ \dot{v} = \dot{r} = 0$ and the ship's speed is reduced by 60% from the speed it had when the turning circle was initiated (Bonilla, 1979).

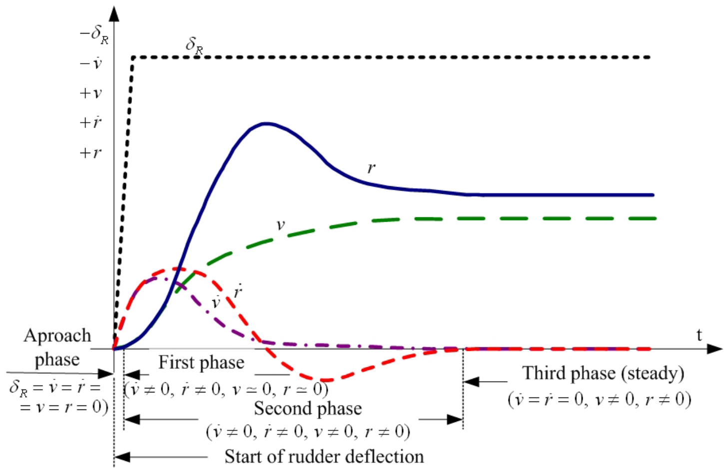


Fig. 2. Characteristics of phases of a turning circle.

2.2 Pull-Out Manoeuvre

The pull-out manoeuvre is a simple test used to obtain a rapid indication of the stability of a straight-line course held by a ship.

A rudder angle of approximately 20º is applied and time is allowed to pass until the ship reaches a constant change of heading rate $r = \dot{\psi}$; at that instant, the rudder is returned to amidships (neutral position). If the ship is stable, the speed will drop to zero both for port and starboard rudder changes. If the ship is unstable, the change of heading rate will drop to some residual speed rate.

This manoeuvre must be performed in both directions, port and starboard, to show any possible asymmetry. Figure 3 shows the results of a pull-out manoeuvre for a stable ship sailing in a straight line and for another unstable one.

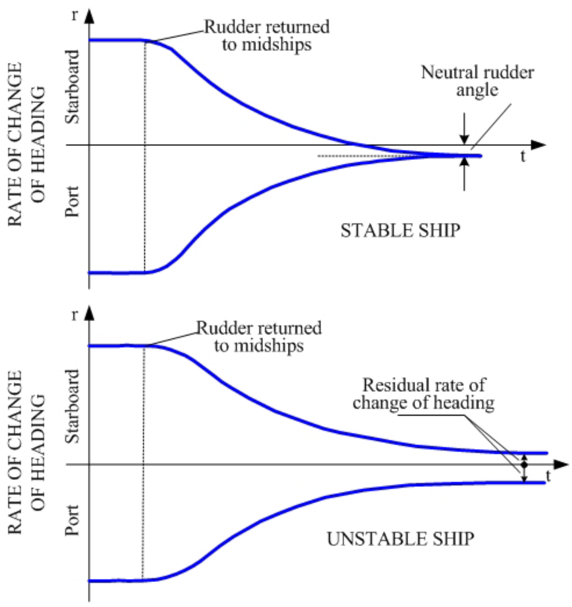


Fig. 3. Presentation of results of the Pull-out manoeuvre.

2.3 Kempf's Zig-Zag Manoeuvre

The zig-zag manoeuvre is obtained by inverting the rudder alternatively by δº to both sides, with a shift of ψ from the initial course. The typical procedure is as follows (Lewis, 1989):

- Make the ship sail in advance and in a straight line at a predetermined speed for a certain time.

- Place the rudder to the starboard side at the maximum speed for a predetermined quantity of δ, for example 10º, and maintain this value until the preselected (10º) course changing ψ occurs (10º) (first operation).
- Place the rudder at the maximum speed on the opposite side (port) at the same angle (10º) (second operation). Maintain the rudder position and the ship continues to rotate in the original direction, at a rotation speed which drops gradually until the movement stops. Then, in response to the rudder, the ship turns to port. The rudder position is held until the preestablished course changing ψº is obtained on the opposite side (port). This completes the overshoot test.
- To complete the zig-zag test, the rudder is again set at the maximum speed at the same angle (10º) on the initial side (starboard) (third execution). Continue until a total of 5 executions of the rudder are completed.

The normal course changing value ψ is 10º. A modified trial can also be taken into account with a course changing of 20º. The 14th ITTC conference recommends executing the manoeuvres at maximum approach speed and, if possible, also at medium speed.

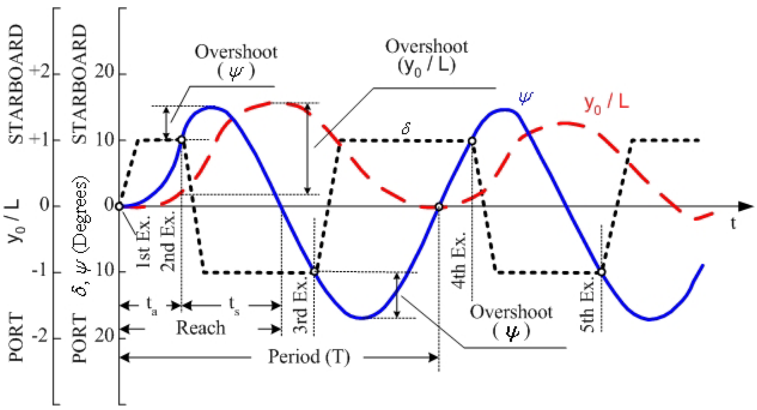


Fig. 4. Zig-zag manoeuvre graph: rudder angle δ , ψ ship's course, yo/L normal distance to the initial trajectory divided by the ship's length.

The results of this manoeuvre are indicators of the capacity of the rudder to control the ship's heading. They can also be used to compare different ship manoeuvring capacities. It should be noted that, from the point of view of the interpretation of the international rules of sea sailing, the use of rudder angles δ to starboard to verify the turning capacity and heading control of a ship are recommended, since, in case of emergency, changes in heading must be made to starboard. For this reason, the normal zig-zag manoeuvre begins with the application of the rudder angle to starboard.

For a simple, initial analysis of the results, the characteristic heading values defined in Figure 4 can be used. The values are given as a function of rudder angle δ.

The main measurements obtained are:

- The time ta it takes to reach the second execution of the heading, which indicates the capacity of the ship to change heading course or the efficiency of the rudder.
- The angle of overshoot in the heading.
- The overshoot of the trajectory obtained when performing the trial.

These latter two measurements are indicative of the amount of anticipation required by the helmsman to sail in restricted waters. In (Arentzen y Mandel, 1960), it is shown that the magnitude of the overshoot in the heading drops when the stability increases but increases when rudder efficiency increases.

The results of the zig-zag test depend on the ship's speed, since the time it takes to reach a given heading falls when this increases.

2.4 Direct and Reverse Spiral Manoeuvres

These manoeuvres provide a qualitative measure of the directional stability of the ship in a straight line. For ships which show stable characteristics, the Dieudonné direct or Bech reverse spiral tests can be used to obtain the response to small rudder angles. For unstable ships, the 14th ITTC recommends the Bech reverse spiral test within the limits indicated by the results of the pull-out manoeuvres.

2.4.1 Dieudonné Direct Spiral Manoeuvre

The direct spiral manoeuvre is used to determine the directional stability characteristics of the vessel, and also provides information on the degree of stability and range of validity of the linear theory.

The procedure for performing this manoeuvre is as follows:

- Make the ship sail in advance with an initial straight course at constant speed.
- Set the rudder at an angle δ, of 25º to starboard and keep it there until the rate of change of heading is constant $r = \dot{\psi}$.
- Once a constant value is reached, the rudder angle δ is reduced by 5º and again held until steady conditions of turning have been obtained.
- This procedure is repeated until the rudder has run through all of the rudder angles from 25º to starboard to 25º to port and again to starboard.

In the range of rudder angles from 5º on either side of zero or neutral rudder angle, the intervals must be reduced.

With this procedure, the ship performs a spiral movement. The graph shows the ship's yaw rate $r = \dot{\psi}$ as a function of each angle δ of the rudder, such as those shown in Figures 5 and 6. This manoeuvre should be carried out in still air and calm water conditions.

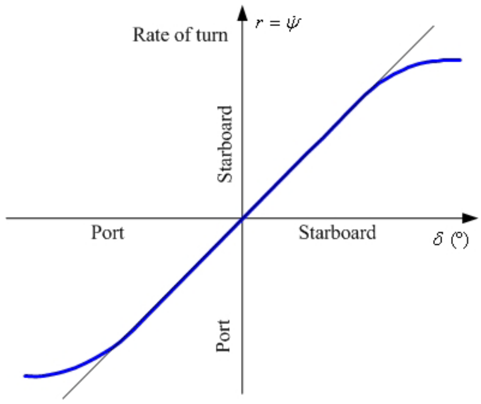


Fig. 5. Graph of direct spiral manoeuvre of a stable, symmetrical vessel.

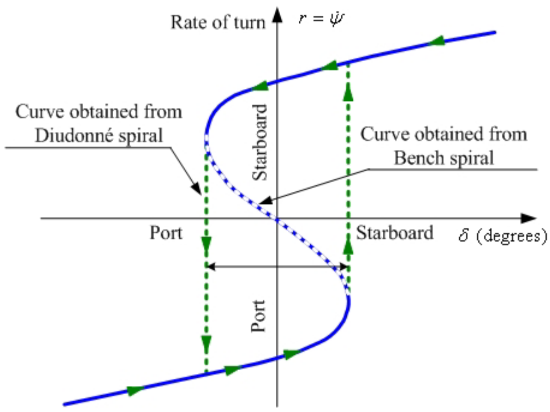


Fig. 6. Graphs of the Dieudonné and Bech spiral manoeuvres for an unstable ship with a hysteresis cycle.

In carrying out the manoeuvre, it is essential to leave sufficient time to reach the stationary state at each angle the rudder is set at. In Lewis (1989), the results of three tests performed with different time intervals between consecutive angles of rudder setting are presented (Strom-Tejsen, 1965). It is shown that to perform an exact study of the stability of a ship, it is essential not to limit the experimental time between the rudder angles.

An indication of the stability of the ship can be obtained from this graph. For example, if it is a single line that goes from starboard to port and is inverse, as shown in Figure 5, then the ship is stable (has stability in a straight course). However, if the graph presents two branches formed by a hysteresis cycle, the ship is unstable. The height and width of the cycle measure the degree of instability, so that the wider the hysteresis cycle, the more unstable the vessel.

2.4.2 Reverse Spiral Manoeuvre

Bech's reverse spiral manoeuvre is an alternative to the direct spiral manoeuvre (Bech, 1968). In this manoeuvre, the ship's course is set at a constant change of heading speed and the rudder angle δ required to produce this rate of turn $r = \dot{\psi}$ is also set. In this trial, the values of the points of the rate of turn curve with respect to the rudder angle can be taken in any order.

The equipment required is a rate-gyro (alternatively, the heading ψ given by the gyro-compass can be differentiated to provide $r = \dot{\psi}$), and an exact indicator of the rudder angle δ. The accuracy of the trial can be improved if the information on the rate of turn and the rudder angle are available continuously. If manual control is used, the helmsman can visualise the instantaneous change of heading speed in a register or indicator. The procedure originally proposed by Bench for obtaining a point in the curve is recommended and is outlined below.

The ship is made to approach the desired change of rate of turn, $r_0 = \dot{\psi}_0$, applying a moderate rudder angle. Once the desired change of heading rate is obtained, the rudder is activated to maintain this desired rate of change of heading as accurately as possible. The helmsman must attempt to maintain the desired change of heading rate using shorter and shorter rudder movements until constant values of the ship's speed and rate of turn are obtained. Normally, a stable change of heading rate is obtained quite fast, so that it is easier to perform the test using a rate-gyro than with a normal gyrocompass.

3 Manoeuvre Models Used

An important decision in this study has been the selection of the type of ships to be used for performing the simulation. An important factor in this choice is the availability of the hydrodynamic derivatives that allow sufficiently reliable dynamic models to be obtained. The ships should preferably belong to a very general class of merchant ships on which there is plentiful data in the literature. Here, a 3 GDL model and another 4 GDL model will be used for which a full dynamic model is available.

3.1 Model of a Mariner Class Ship

As a model example of a ship with 3 GDL, the model of a Mariner class ship, widely used in the literature, has been selected. Data from the ship 'USS Compass Island' (Chiselett y Strom-Tejsen, 1965) have been used. The main characteristics of the ship are shown in Table 1.

Table 1. Main dimensions of the Mariner class ship

Description	Symbol	Value	Units
Length between perpendiculars	L_{pp}	160,93	m
Maximun beam	B	23,17	m
Design draft	D	8,23	m
Design displacement	∇	18541	m3
Design speed	U_0	20	knots
Max. design rudder angle	δ	40	deg
Max. design rudder rate	$\dot{\delta}_{\max}$	2,5 – 3,7	deg/sec

The mathematical model used for the simulation with three degrees of freedom is:

$$\begin{bmatrix} m' - X'_{\dot{x}} & 0 & 0 \\ 0 & m' - Y'_{\dot{v}} & m'x'_G - Y'_{\dot{r}} \\ 0 & m'x'_G - N'_{\dot{v}} & I'_z - N'_{\dot{r}} \end{bmatrix} \begin{bmatrix} \dot{u}' \\ \dot{v}' \\ \dot{r}' \end{bmatrix} = \begin{bmatrix} X' \\ Y' \\ N' \end{bmatrix} \quad (1)$$

The non-linear forces X and Y and the hydrodynamic moment N are developed using the Abkowitz (1964) model.

To develop the simulation model, the cinematic equations must be added, giving:

$$\mathbf{M}\,\dot{\mathbf{v}}' = \boldsymbol{\tau}' \quad (2)$$

$$\dot{\boldsymbol{\eta}} = \mathbf{J}(\boldsymbol{\eta})\mathbf{v}' \quad (3)$$

To include the model of the rudder action, the simplified model suggested by Van Amerongen (1982) has been used, as indicated in Figure 7 where δ_c is the rudder angle demanded by the controller and δ is the real rudder angle. The typical saturation values of the rudder angle and turning speed are in the following value ranges:

$$\delta_{\max} = 35^{\circ} \text{ and } 2.5^{\circ}/seg \leq \dot{\delta}_{\max} < 7^{\circ}/seg.$$

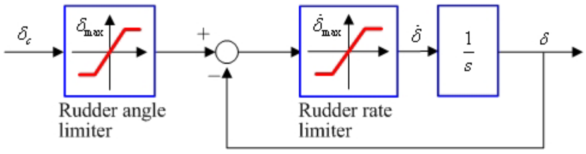


Fig. 7. Simplified Diagram of the rudder control loop.

In the development of this model, in accordance with the characteristics of the Mariner-type vessel, the rudder angle and rudder rate limits have been set at 30º and 4,6 /s respectively.

3.1.1 Description of the Simulink Model

The mathematical model has been developed in the Matlab-Simulink environment. Figure 8 shows the block diagram used. The main block receives as input the desired rudder angle δc and generates as output the ship's heading angle ψ . The model of the ship has been separated into two blocks. The first contains the rudder behaviour model and the second the ship dynamics model.

The ship dynamics model has been developed using an s function of Matlab which receives as input the rudder value δ and generates as output: the longitudinal advance speed u, the transversal speed v, the turning speed r, the heading ψ, the ship's position x, y, the rudder angle δ and the ship's speed U.

3.1.2 Sea Trials Simulation

Firstly, Figure 9 shows a graph of the rudder dynamics. It can be observed that a time of approximately 10 s is required to go from the neutral position of 0º to the maximum angle allowed (± 30º) and 19 s to make the maximum change from -30º to 30º which is a normal result in this type of vessels.

The turning circle, zig-zag and spiral tests have also been simulated. The results of the simulations can be used to obtain an initial estimate of the dynamics behaviour and the stability and manoeuvrability characteristics of the ship.

Figure 10 shows the turning circle for the rudder angles 5º, 10º, 15º, 20º and 25º. It can be observed that the gyro radius drops as the rudder angle δ increases.

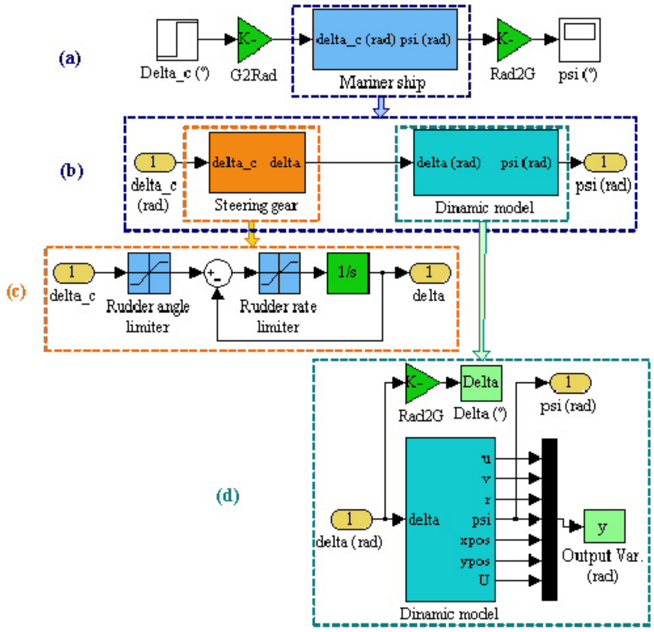


Fig. 8. Simulink block diagrams used in simulation: (a) Main block of ship (b) Separation into two blocks, steering gear and dynamic model. (c) Diagram of model used in steering gear. (d) Block diagrams of dynamics model indicating output variables.

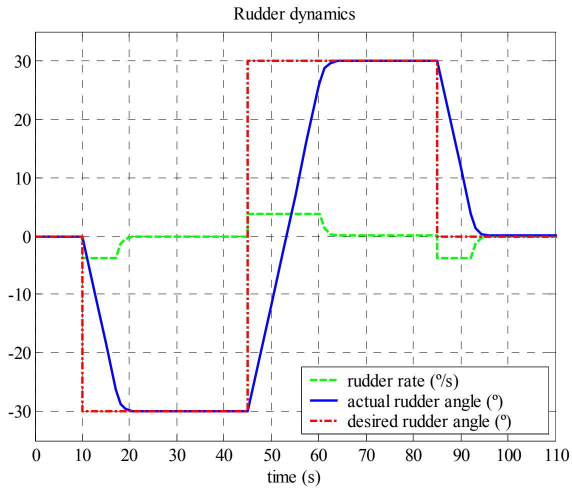


Fig. 9. Rudder dynamics behavior.

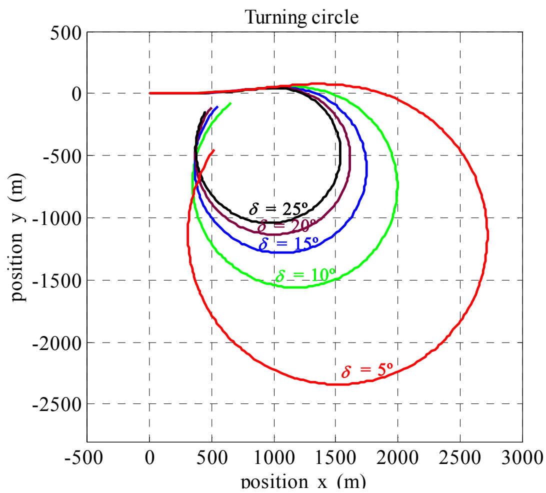


Fig. 10. Turning circle for various rudder angles δ.

Figure 11 shows the results of a zig-zag manoeuvre 20º/20º. The results of this trial are indicative of the manoeuvring capacity of the ship for a specified rudder angle.

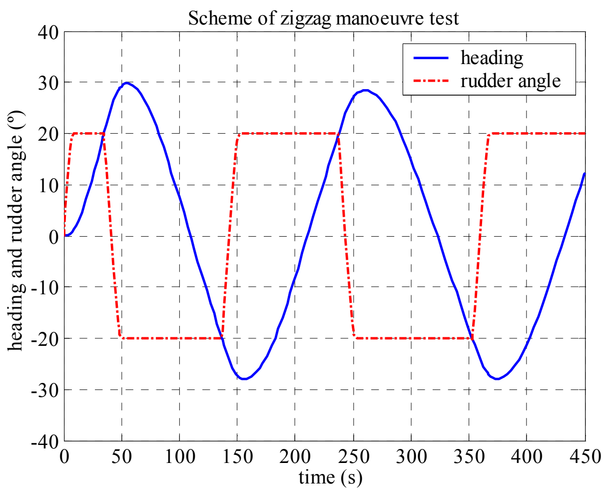


Fig. 11. Zig-zag test 20º-20º.

The simulation of the spiral manoeuver is shown in Figure 12, representing the reduction speed of the ship $r = \dot{\psi}$ as a function of each angle δ set at the rudder. The graph obtained indicates that the ship has a stable behavior on a straight course. The slope of the line tangent to the curve allows us to determine the gain in the linear approach over the Nomoto model.

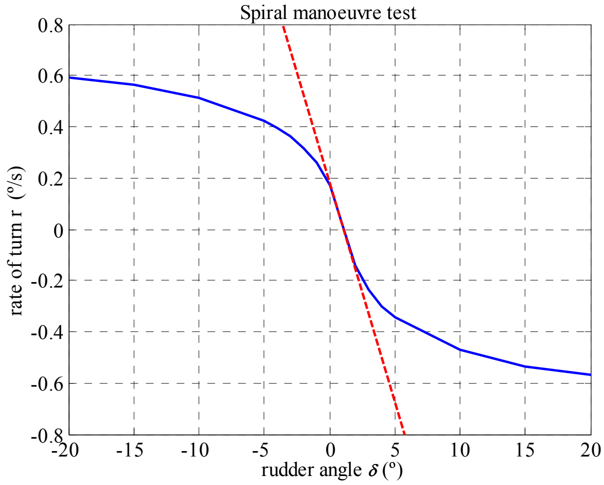


Fig. 12. Graph with results of spiral test 20º/20º.

A stable behavior of the ship is also observed in the results of the pull-out manoeuvre as can be seen in Figure 13.

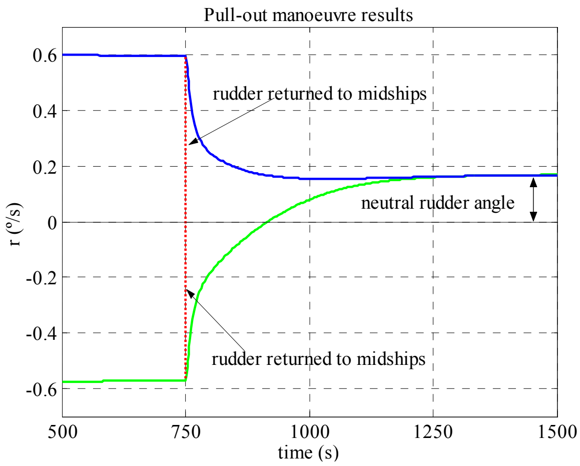


Fig. 13. Pull-out manoeuvre

3.2 Son and Nomoto Model

As a model example of a ship with 4 GDL, the simulation model of the SR 108 container ship that Son and Nomoto (1982) identified in their research has been developed. The main characteristics of the ship are shown in Table 2.

Table 2. Main dimensions of the SR 108 container ship

Description	Symbol	Value	Units
Length between perpendiculars	L	175.00	m
Beam	B	25.40	m
Displacement	∇	21.222	m^3
Ship's speed	U_0	32,2	knots
Nominal propeller	A_R	158	rpm

The mathematical model used for the simulation with four degrees of freedom is:

$$\begin{bmatrix} (m'+m'_x) & 0 & 0 & 0 \\ 0 & (m'+m'_y) & -m'_y l'_y & m'_y \alpha'_y \\ 0 & -m'_y I'_y & (I'_x+J'_x) & 0 \\ 0 & m'_y\alpha'_y & 0 & (I'_z+J'_z) \end{bmatrix} \begin{bmatrix} \dot{u}' \\ \dot{v}' \\ \dot{p}' \\ \dot{r}' \end{bmatrix} = \begin{bmatrix} X' \\ Y' \\ K' \\ N' \end{bmatrix} + \begin{bmatrix} (m'+m'_y)v'r' \\ (m'+m'_x)u'r' \\ m'_x l'_x u'r' - W'GM'\phi' \\ -Y'x'_G \end{bmatrix} \quad (4)$$

where all the equations are in the adimensional form, in the "Prima" system. The variables m'_x y m'_y are masses added in the directions *x* and *y* (advance and lateral displacement), J'_x y J'_y are momentums of inertia added around the *z* and *x* axes (yaw and balance). The coordinate *x* of the centre of the mass m_y is α_y, while l'_x y l'_y are the coordinates *z* of the centres of the added masses m_x and m_y respectively.

The origin of the fixed coordinates of the ship is decribed as $R_G = (x_G, 0, 0)$. The terms of the added masses and inertias are expressly included with their corresponding gyro radii. It also includes the restoring moment *W'GZ'*(ϕ) in its approximate form, *W'GM'*ϕ*'*. The term x'_G appears in the equation since the hydrodynamic momentum of the yaw *N'* is defined around the geometrical centre of the ship.

Grouping together the terms of the second member of equation (4), as indicated in equation (5):

$$\boldsymbol{\tau}' = \begin{bmatrix} X'_1 \\ Y'_1 \\ K'_1 \\ N'_1 \end{bmatrix} = \begin{bmatrix} X' \\ Y' \\ K' \\ N' \end{bmatrix} + \begin{bmatrix} (m'+m'_y)v'r' \\ (m'+m'_x)u'r' \\ m'_x l'_x r' - W'GM'\phi' \\ -Y'x'_G \end{bmatrix} \quad (5)$$

Thus, the equations for the movement of the model can also be expressed in vector form as:

$$\mathbf{M}\,\dot{\mathbf{v}}' = \boldsymbol{\tau}' \tag{6}$$

$$\dot{\boldsymbol{\eta}} = \mathbf{J}(\boldsymbol{\eta})\mathbf{v}' \tag{7}$$

3.2.1 Rudder Dynamics and Saturation

To include the rudder action in the model, the simplified model proposed by Van Amerongen (1982) has been used, as shown in Figure 7.

In the development of this model, in accordance with the characteristics of the container ship, the rudder angle and speed limits are set at 20º and 3º/s, respectively.

3.2.2 Dynamics and Saturation of Propeller Shaft

The propeller dynamics used in this model is the first order system shown in Figure 14.

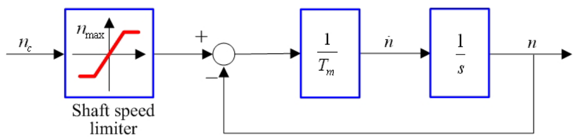


Fig. 14. Propeller shaft model

The behaviour of the system depends on the real value of the rudder speed. There are two operating modes which determine the value of the T_m parameter: one for propeller axis gyro speed values of $n \leq 20$ rpm with a constant value of T_m =18,83, and the other for axis values of $n > 20$ rpm with a T_m value that depends on n, defined by the equation T_m =5,65/(n/60).

3.2.3 Description of Simulink Model

The mathematical model has also been developed in the Matlab-Simulink environment. Figure 15 shows the block diagram used. In this model, the main block includes two inputs: the angle δ_c, the desired rudder angle in radians and the propeller axis gyro speed in rpm, and generates as output the angle ψ of the ship's course also in radians.

As well as the block used to represent the rudder behaviour, another block has been separately incorporated to represent the behaviour of the propeller axis. The propeller axis dynamics model has been developed in accordance with values and characteristics outlined in the section above.

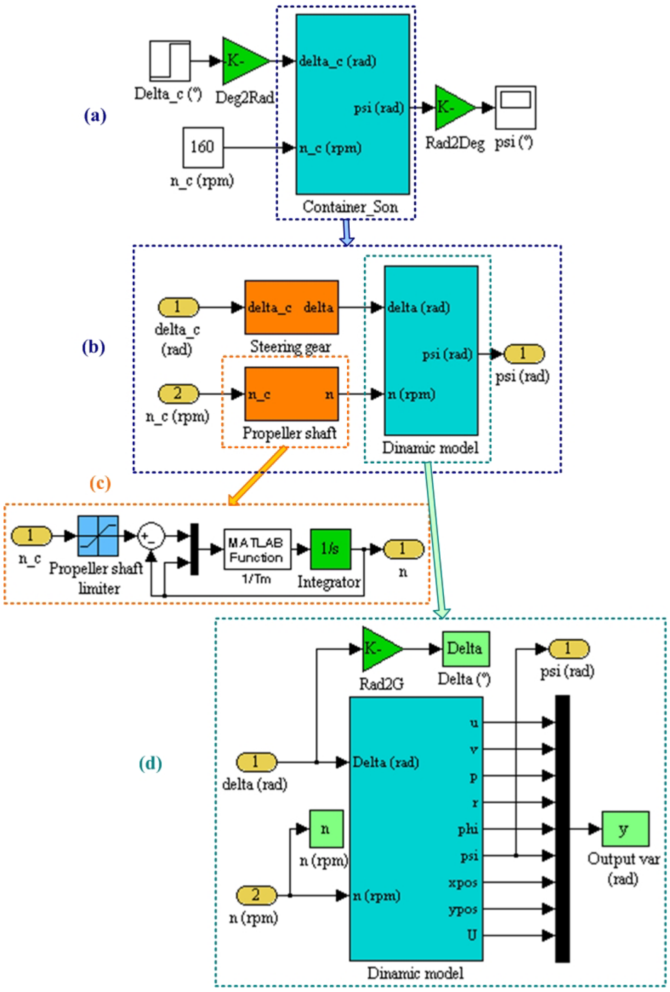


Fig.15. Simulink block diagrams used in the simulation: (a) Main block of ship (b) Separation of steering gear, propeller shaft and dynamic model blocks, (c) Diagram of model used for propeller shaft (d) Dynamic model block diagram indicating output variables

The ship dynamics model has been developed using an s function of Matlab which receives as input the rudder value δ and the propeller speed n and generates as output: the longitudinal advance velocity u, the transversal velocity v, the turning velocity r, the roll velocity p, the heading ψ, the roll angle ϕ, the ship's position x, y, the rudder angle δ and the ship's speed U.

With this model, manoeuvres can be simulated at different propeller speeds, up to the threshold value of n = 158,19 rpm such as, for example: Low-speed ahead = 40 rpm, Médium speed ahead = 80 rpm, full manoeuvre speed ahead = 120 rpm and full free sailing speed ahead = 158 rpm, as shown in Table 3. This table also shows the ship's speed and the time required to reach a constant speed for each of the propeller speeds. However, it has the limitation that it only allows advance sailing commands.

Table 3. Effect of propeller speed variation

Command to the engine	Propeller speed (rpm)	Ship speed (m/s)	Ship speed (knots)	Time (s)
Low ahead	40	3,18	8,1	1.300
Medium ahead	80	8,37	16,2	750
Full ahead (manoeuvre)	120	12,56	23,4	550
Full ahead (sailing)	158	16,56	32,2	440

It is also possible to simulate the ship's behaviour for GM metacentric height values from GM = 0.45 m to GM = 3 m.

3.2.4 Sea Trial Simulations

First, the dynamic behaviour of the propeller is simulated, together with its acceleration and deceleration capacity, as shown in Figure 16.

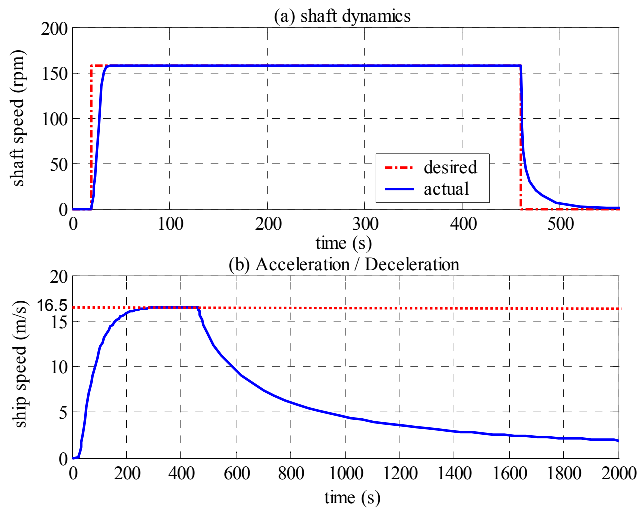


Fig. 16. Dynamic behaviour of the propeller shaft and acceleration and deceleration capacity

In Figure 16.a, it can be observed that the propeller shaft needs 24s to reach the maximum speed and approximately 100 s to stop. Figure 16.b, shows that the time required to pass from zero speed to maximum full-ahead sailing speed is 440 s,, and that of the reduction in speed from full-ahead sailing speed to zero is over 2000 s.

Figure 17 shows a graph of the dynamic behaviour of the rudder. It can be observed that it takes approximately 10 s to pass from the neutral position of 0º to the maximum value (± 20º) and 17 s to make the change from -20º to +20º.

Figures 18 (a) and (b), show the results of the three evolution tests performed with an initial speed of 12,56 m/s for a rudder angle δ of 15º. The tests performed for the GM metacentric height values of 0,3 m, 0,5 m and 3 m, including the roll angle graphs. It can be observed that when the GM metacentric height drops, the ship's gyro radius falls, while the roll angle increases considerably.

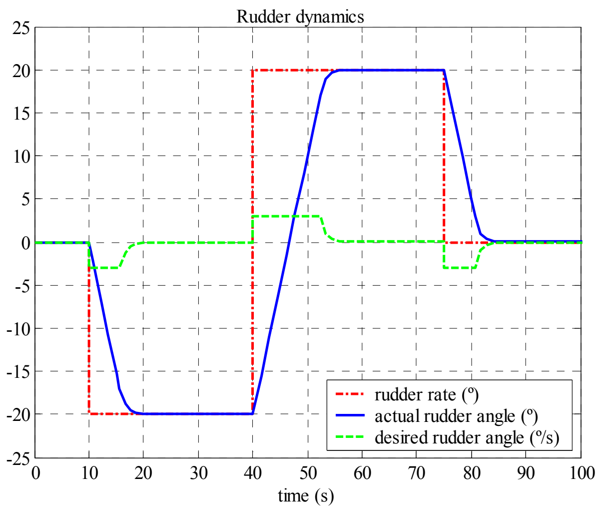


Fig. 17. Dynamic behaviour of the rudder

Figures 19 (a) and (b), present the results of a 20º/20º zig-zag manoeuvre for the threshold values of the GM metacentric height of 0,3 m and 3 m, including the balance evolution graphs during the trial. It can also be observed here that the balance angle increases when the GM metacentric height diminishes.

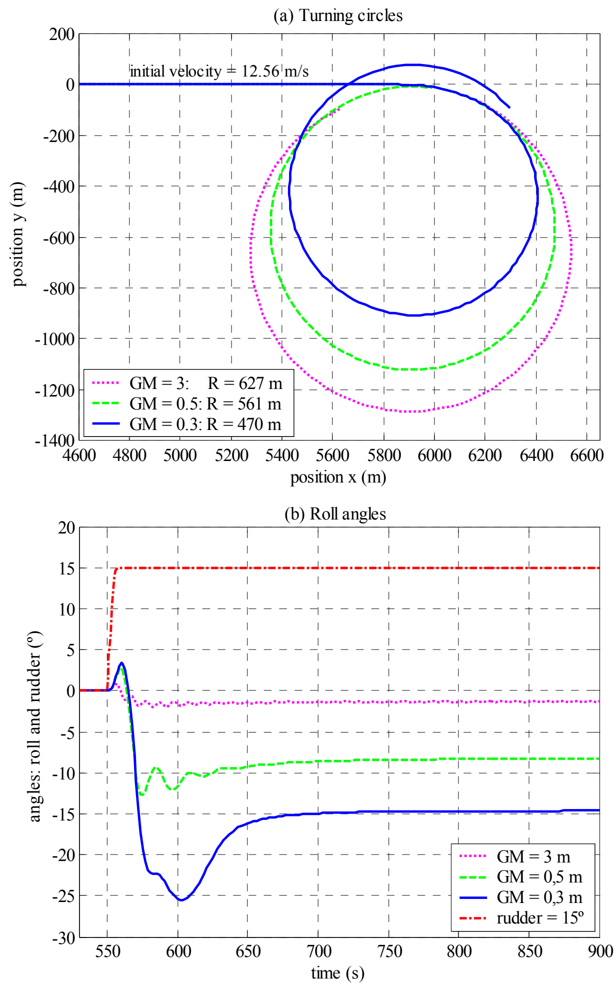


Fig. 18. Turning circle manoeuvres performed with the propeller at 120 rpm (ship's speed of 12.56 m/s) with a rudder angle δ of 15° and GM values of 0,3 m, 0,5 m and 3 m: a) turning circles; b) roll angles

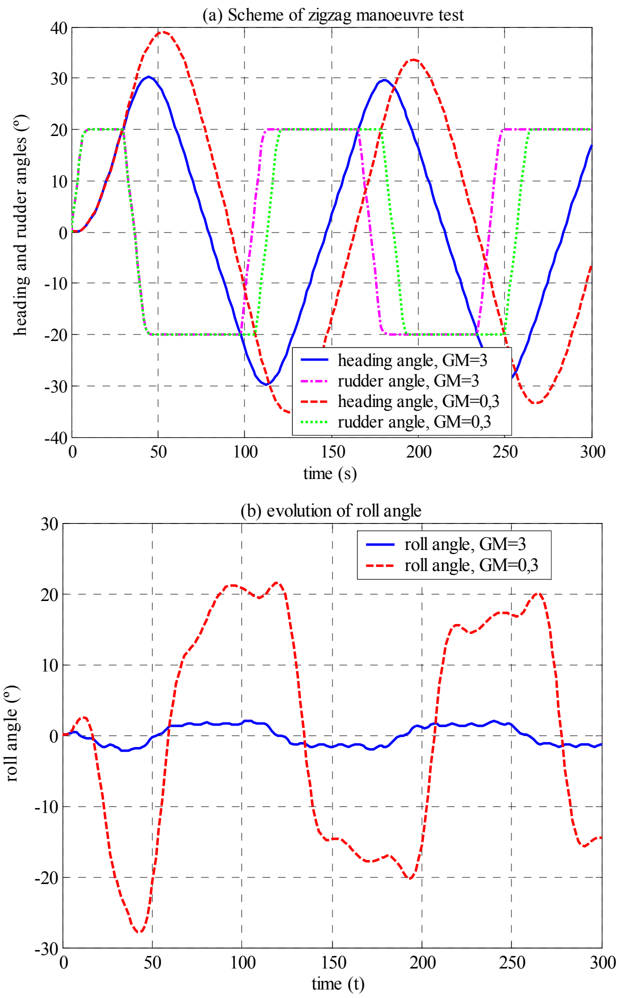


Fig. 19. 20º/20º zig-zag manoeuvre for GM values of 0,3 m and 1 m. a) evolution of heading and rudder angle. b) evolution of roll angle

Conclusions

Although sea trials are carried out with the ship already built and at sea, it is useful to simulate the dynamic behaviour of the ship in the design stage using mathematical models.

Some of the most widely used sea trials have been simulated using two mathematical models with three and four degrees of freedom. The simulation results allow the ship's dynamic behaviour characteristics to be measured, an indication of its stability on a straight course to be obtained, its robustness and the limitations of the control system to be evaluated and its behaviour in emergency situations to be assessed.

The results obtained in the sea trial simulations indicate that sea trial simulation tools form an important part of the design stage in the building of a ship.

Acknowledgements

This work has been partially financed by the MCYT through project DPI2003-09745-C04-03.

References

14th ITTC. 1975. Discussions and Recommendations for an ITTC 1975 Maneuvering Trial Code. 14th International Towing Tank conference, Ottawa, September 1975, pp. 348-365.

Abkowitz M.A. 1964. Lectures on Ship Hydrodynamics, Steering and Manoeuvrability, Technical Report Hy-5 Hydrodynamics Department, hydro and Aerodynamics Laboratory, Lyngby, Denmark.

Amerongen J. van 1982. Adaptive steering of ships – a model reference approach to improved manoeuvring and economical course keeping. Thesis (PhD). Delft University of Technology.

Arentzen, E.S., and Mandel, P. 1960. Naval Architecture Aspects of Submarine Design. SNAME Transactions. 68.

Bech, M.I. 1968. The reverse Spiral Test as Applied to Large Ships, Shipping World and Shipbuilder, pp.1753-1754.

Bonilla de la Corte A. 1979. Teoría del buque. Librería San José. Vigo.

Chislett, M.S. and Støm-Tejsen, J. 1965. Planar Motion Mechanism Tests and full-scale Steering and Maneuvering Predictions for a Mariner Class Vessel. Technical Report Hy-6. Hydrodynamics Department, Hydro and Aerodynamics Laboratory, Lyngby, Denmark.

Haro M. 2004. A new Recursive Identification Procedure of the Nonlinear Model Ship Based on the Turning Test Manoeuvring and the Norrbin Equation. Journal of Maritime Research. Vol. 1, Nº 2.

Lewis E.V. 1989. Principles of Naval Architecture. (2nd ed.), Vol. 3. The Society of Naval Architects and Marine Engineers

CHAPTER 12

Platform for remote experimentation of autonomous high speed craft models with a combined technology for wifi and internet communications programmed in LabVIEW

F. J. VELASCO[1], T. M. RUEDA[1], E. REVESTIDO[1], E. MOYANO[2], E. LOPEZ[3] and L. A. ESQUIBEL[4]

[1]Dpto. Tecnología Electrónica e Ingeniería de Sistemas y Automática, Univ. de Cantabria. velascof@unican.es. ruedat@unican.es; revestidoe@unican.es
[2]Dpto. de Matemática Aplicada y Ciencias de la Computación. Univ. de Cantabria. moyanoe@unican.es
[3]Dpto de Ciencias y Técnicas de la Navegación, Máquinas y Construcciones Navales, Universidad del País Vasco (UPV/EHU). cnplogae@ehu.es
[4]Dpto. Ingeniería Eléctrica y Electrónica. Univ. del León. luisangel@unileon.es;

This chapter describes an experimentation environment for the various tests and manoeuvres for testing the stability and steerability of sea vessels with autonomous in-scale physical models. The model has an Industrial PC which communicates through a wireless network with the laptop on land, which can in turn be connected to other PCs through Internet using 3G UMTS technology. A software support has been implemented in the Industrial PC, which is capable of acquiring and storing data from all of the test platform instruments from a distance. It is possible from a distance, via the web and via DataSocket, to view data and modify the parameters of all of the instruments of the platform using the wireless network with wifi technology and also the Internet network with 3G UMTS technology. To do this, a software support is used, developed in LabVIEW and suitable for carrying out the Guidance, Navigation and Control of the physical model. This model makes it possible to perform the sea trials most widely used to

determine the main characteristics of the steering and manoeuvring of a sea vessel, such as: evolution curve, zig-zag manoeuvre, pull-out manoeuvre and spiral manoeuvre.

1 Introduction

Within the framework of the project Automatisation of Marine Vehicles for Cooperative Actions (Environment for Cooperation of Marine Vehicles) DPI2003-09745-C04-03, one of the main lines of research is the study of a ship's manoeuvres for actions in cooperation with other vehicles, such as aiding a damaged ship, tow manoeuvre and joint actions or sailing in close proximity. The aim of the project is to design controllers for the control of these manoeuvres and for the following of pre-planned trajectories. Experiments are also carried out in an uncontrolled medium, in the form of sea trials, for the validation of the designs made, as a prior step to their actual implementation.

This chapter presents the description of a test platform which allows data gathering and the steering control of an autonomous in-scale physical model of a high-speed ship, TF-120 (Figure 1), for the Guidance, navigation and Control (GNC) (Fossen, 1994) of this model from a distance. With this platform, the various tasks of coordination between marine vehicles can be performed, such as towing tasks, for cooperative actions.

Fig. 1. Turbo Ferry TF-120

In order to obtain a good estimate of the position, speed, acceleration and heading of the vehicle and thus to enable the vehicle to follow a preestablished trajectory (Hansen, 1996) with the minimum error possible despite possible outside disturbances such as currents, winds, tides and obstacles, a sensorial integration must be carried out. This platform is also designed to perform the tests necessary for the selection of the optimal trajectory following strategy.

All of the elements that make up the system, which will be described in the following sections, are fan industrial type. The main reason for this choice is the robustness and reliability provided by these elements. With this test environment, the installations of a vessel are efficiently emulated. The tests with the autonomous in-scale physical model of the high-speed ship were carried out in the surroundings of the Bay of Santander.

The software application designed is capable of making all of the calculations required to plan the trajectories using the input data, which are: starting point, final point, type of trajectory and time invested in finding this trajectory. With these calculations, the programme generates the references required to make the dynamic controls (Ollero, 2001) of the physical model, using the control strategy considered most appropriate.

2 Elements that Make up the Environment

Figure 2 shows all of the components that form part of the remote experimentation environment for marine vehicles. This environment is divided into two parts: one includes the elements on board and the other the elements which are on land. The communication between the two parts is made through a wireless system with two points of access: one on land connected via a wireless system to a laptop and another for the marine vehicle connected to the Industrial PC through network cable. The laptop also has a 3G UMTS card, which allows another PC or PCs to connect through Internet to the laptop.

The elements on board are as follows: the propulsion and control elements, the control electronics, the receiver station, the instrumentation, the Industrial PC and one access point. The elements on land are: the laptop with the possibility of connecting to another or other PCs through Internet, the access point and the transmitter station.

2.1 Instrumentation of the Environment

The environment has a KVH (Technical Manual KVH) electronic gyrocompass which measures: the current heading, position, time, etc. This data is acquired thanks to the RS232 serial gate of the PC through the NMEA 0813 (National Marine Electronics Association) communication protocol. The data is stored in file to perform an identification (Ameron-

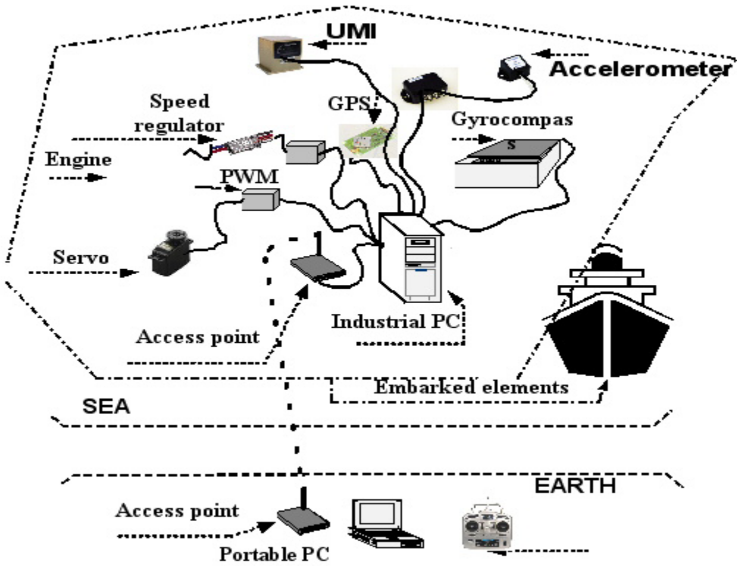


Fig. 2. Elements required for following a trajectory.

gen, 1982) of the mathematical model that relates the current heading with the propulsion gyro angle.

Using the GPS RCB-LJ receiver which incorporates the TIM-LF (u-blox, 2004) chip, fairly accurate data on position and speed, among other parameters, are obtained. An electric circuit had to be designed to perform the conversion from CMOS to TTL levels and which also incorporates an RS232 serial gate adaptor. In this way, the GPS receiver can send and receive data using the communication protocol selected. the TIM-LF chip admits the protocols UBX, NMEA Y RTCM.

To obtain data on the movement of the physical model, two triaxial accelerometers are used, one placed at bow and one at stern. The inertial measurement unit also provides the values of the six degrees of freedom of the movement of the model. This data is necessary for establishing the vertical movements of the model.

The disposition of the elements on the vessel is intended to effect an adequate distribution of weights, apart from the necessary ballast. Also with this distribution electromagnetic compatibility problems are avoided. The distance between the electronic gyrocompass and the other elements must be over 1 metre and, in this case, the nearest ones are the batteries and the electric cupboard. The gyrocompass is sensitive to adjacent magnetic fields, with the possibility of the appearance of magnetic alterations impossible to redress. The wireless access points may be affected by interferences produced by other elements working in the band of 2.4 GHz. The GPS antenna is also susceptible to interferences produced by the point of access antenna making it necessary to place this at a safe distance from the antenna of this point of access.

2.2 Actuators and Control Electronics

The propulsion is made up of a series of motors which make the turbine rotate around their corresponding motorjets. In order to emulate the propulsion of a real ship, some fixed gyro angle turbojets have been fitted in the central part of stern of the model and at each side of these some variable angle turbojets. The motors are regulated by a speed variator, which is commanded by a PWM (pulse width modulation) circuit control. For the correct functioning of the speed variators and the Futaba motor control servos, it is necessary to generate a PWM control signal of around 55,6 Hz. For this reason, it was necessary to design a custom-built PWM circuit, using 555 integrals. The direction of the turbojet propeller fluid is controled with a servo motor, which is in turn controlled by a PWM control circuit.

An application called SCADA_Industrial.vi installed in the Industrial PC will be in charge of presenting on screen and storing all the data measured by the instruments on board. It is also possible to modify the control parameters to follow a given trajectory. LabVIEW (Bishop, 2004) allows communication between the SCADA_Laptop.vi application being run on the laptop PC and its corresponding application in the Industrial PC (SCADA_Industrial.vi) using different communication protocols. The exchange of data between the various PCs is possible thanks to the publication of data with a DataSocket server or a web server.

The SCADA_Laptop.vi application Publishes data on the DataSocket server to be able to exchange data with the corresponding application on the Industrial PC (SCADA_Industrial.vi) and also publishes data on the web servers that data can be exchanged with another or other computers. That is, from a computer connected to Internet, the application of the laptop PC can be accessed with the right address and this application of the laptop can access the application of the industrial PC thanks to the DataSocket server.

The real-time calculation and data-gathering power are found in the Industrial PC, which is equipped to support this requirement.

The elements described up to now form part of the system in automatic mode, but the system also has a manual mode, through which it is possible to manage the marine vehicle by means of a commercial modelling radio control. The signal sent by radio control is received by a receiver station installed on the marine vehicle. The receiver station has 4 channels which are: a PWM control signal for controlling the motor direction servos, another for the speed variators, another channel for a relay with which it is possible to pass from automatic to manual mode by means of the radio control remote control, and the fourth for the inversion of the course direction.

The switch from one mode to another is performed by means of a series of industrial relays, which also sirve to separate the power part from the control part. This provides security in the worst possible operating conditions. The normal state of the system corresponds to the manual mode which the system remains in even when the electricity supply is lost, until a signal is sent by the PC to change to automatic mode. Once in automatic mode, the system management is passed on the industrial PC and the system remains in this state until a signal is sent either by the PC or by the radio control handset to switch from automatic to manual mode. The radio control handset signal to switch from automatic to manual mode has prevalence over all others.

3 Development of Software Support

For the development of the software support, version 7.1 of LabVIEW (LabVIEW 7 Express User Manual, 2003) has been selected as the graphic programming environment, as this is an industry Standard and because it enables the simple undertaking of graphic interfaces in real time. Another important characteristic of LabVIEW is that it allows the testing of different types of controllers and different types of communication protocols.

Figure 3 shows the different control signals on the system actuators, such as the propulsion speed control loop, the propulsion direction control loop and the stabiliser flan loops; the bow flap (T-Foil) and the stern flaps (Flaps). Each motor is controlled by a speed variator, a servomotor and a PWM circuit control. The analogue control signal from the data acquisition card can be the same for both motors, or there can be one for each motor. Thus, the gyro angle of the turbojet and the gyro speed of the engine are practically the same.

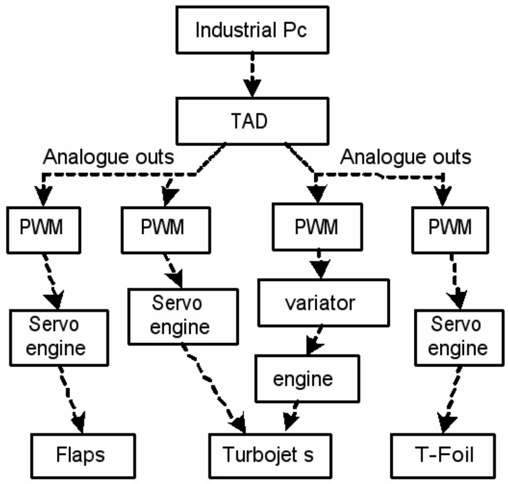


Fig. 3. Propulsion Control

In this software support, all of the manoeuvres are programmed so that they can be performed automatically. To do this, the Industrial PC is equipped with an application called SCADA_Industrial.vi, which, apart from performing the manoeuvres, deals with the gathering and storage of the data from all of the instruments which make up the platform. To fulfil this mission, this process has a series of threads. Thread number 1 fulfils the task of taking and storing data from the electronic gyrocompass.

Thread number 2 forms a heading control loop, using the most appropriate control law, such as a PID. In this heading control thread, it is possible to modify the gyro angle of the turbojets manually or automatically depending on the manoeuvre selected by the user on the graphic interface.

Thread number 3 deals with the gathering of data at the GPS receiver, also using the Standard NMEA, with the following configuration data: 9600 Baudios, 8 bits of data without parity and one stop bit. The most interesting data from the chains of characters sent are position and speed. These data are used by the trajectory planner in order to monitor the trajectory as precisely as possible.

The last thread, number 4, performs a speed control algorithm with the data gathered by the GPS.

When designing these threads, the real time needs of the system have been taken into account. High-level VIs (Virtual Instruments) have not been used as these consume time and resources, which is inadequate for the correct functioning of the system. This is why an appropriate support has been designed to make an efficient use of the PC and to avoid the loss of data.

In order to be able to configure the serial gate correctly using LabVIEW, the VISA 3.0 installation needs to be installed. Among other tasks, this application detects the number of PC serial gates and allows them to be configured inside the application designed in LabVIEW. In order to support the NMEA 0813 protocol, the serial gate is configured with the following parameters: speed 4800 Baudios, 8 bits of data without parity and one stop bit. It was necessary to make an error-sifting algorithm as the character chains reach the computer with a host of errors. The fact of sifting all of the data readings slows down the data acquisition process, so a compromise is made between speed of execution and reliability.

3.1 Software Support Communications System

The communication between the computer located on land and the one positioned on the sea vessel is made through a wireless network using WiFi (Wireless Fidelity) technology based on the standard 802.11g (Planet technology Corp, 2004). The laptop PC can in turn communicate with one or several PCs equipped with a 3G UMTS (3 generation Universal Mobile Telephone System) mobile telephone card with transmission speeds of 384 Kbps.

The connection will function at the maximum speed allowed for maintaining an optimum transmission automatically. This is, for an 802.11g protocol: 54, 48, 36, 24, 18, 12, 9 or 6 Mbps. The speed will depend on the distance the client is from the point of access, on whether or not there is encrypting between the client and the point of access, on the existence of interference in the 2,4 Ghz band (mobile telephones, microwaves, …).

The laptop links with the industrial PC through the access points by means of omnidirectional antennas which allow a distance between access points of around 250 metres, the result being the same as if the two pieces of equipment had been connected to a typical local area network. By having an access point in repeater mode, the coverage is tripled, with the only drawback that the technical specification forbids WPA encrypting between two access points configured in repeater mode, so that all of the information we send can only be encrypted in WEP (Wired Equivalent Privacy) mode, an encryptation which has proven to be vulnerable, even in its 128 bit version. The laptop PC also has a 3G PCMCIA card, which allows access to Internet. Once connected to Internet, it is possible to connect with any other PC or PCs connected to Internet. These PCs may be connected to

Internet via an Ethernet cable or using a 3G PCMCIA card, as can be observed in Figure 4.

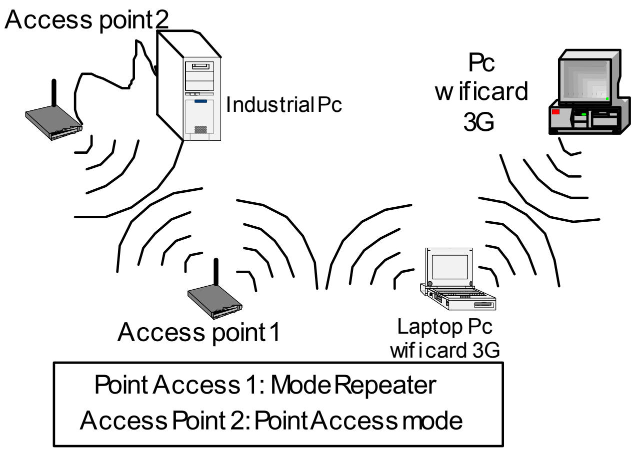


Fig. 4. Elements that make up the wireless network

The original antenna of the access point located on the ship has been changed for a longer omnidirectional one with improved characteristics. It should also be pointed out that the power of the radio control antenna covers a similar distance of communication.

The decision to use this wireless network configuration and not to select one single access point and a wireless network card, is based on grounds of improving security in the communication and extending the reach of the wireless signal.

Once it is verified that the network is properly configured and that there is communication between the various elements described above, the software system can be designed without the risk of encountering communication problems.

3.2 Publication of Vis in DataSocket Server and in Web Server

In order to be able to access the SCADA type application being run on the Industrial PC via the wireless network and at the same time be able to access this application from another or other PCs through Internet, it was necessary to combine DataSocket technology and the web publication of LabVIEW,in the software support, as can be seen in Figure 5.

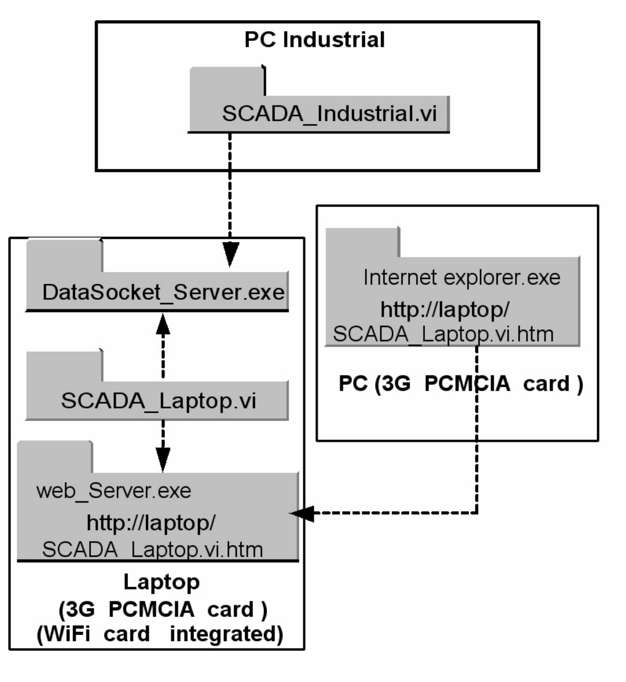


Fig. 5. Software System Applications

There are two graphic interfaces with the same appearance: once called SCADA_Industrial.vi which is run on the Industrial PC and another called SCADA_Laptop.vi which is run on the laptop. These two applications interchange information bidirectionally through the LabVIEW DataSocket server in the laptop. Any control modified by the use in the remote application of the laptop will automatically be modified in the application of the Industrial PC. At the same time, the SCADA_Laptop.vi application publishes its data on the web server which is also in the laptop. Thus, any other PC connected to Internet may have access to this application and take over the control at any moment. A PC, or more than one PC, can be connected through Internet, as can be observed in Figure 5. It must be borne in mind, however, that only one of them can assume control of the application while the others can only visualise the data.

The implemention of the communications system described above might have been undertaken using only DataSocket technology (NI DataSocket Server Help, 2003). However, apart form the SCADA_Industrial.vi for the Industrial PC and SCADA_Laptop.vi for the laptop, it would have been necessary to design one application for each computer connected to Internet that wanted to connect to the laptop. This is why web server technology is also being used since, in this way, any computer connected to Internet can access the laptop by typing in the corresponding address in a search engine, such as Internet Explorer, without the need to create a new application for each computer that wants to connect to the laptop. For all of these reasons, the most adequate solution is to combine DataSocket technology with the labVIEW web server technology.

When using DataSocket technology, each of the controls of the SCADA_Industrial.vi application or the SCADA_Laptop.vi application which form the user interface, such as the scroll bars, on/off buttons, text frames, etc. are the elements denominated 'ítems'.

These ítems are published through a DataSocket Server capable of publiching data so that other client processes can read them or write them. In our system, the SCADA_Laptop.vi application is connected to the DataSocket server where all of the items of this application are published, in reading and writing mode (Figure 6). In the same way, the SCADA_Industrial.vi application is connected to the DataSocket server through the wireless network to access the data published by the client application of the laptop. The Client application of the Industrial PC subscribes to the data published by the laptop and only has reading capacity, not writing.

The DataSocket server shows on screen the numebr of client processes connected, in this case two, the client application of the industrial and laptop computers and the number of packages sent and received. It also has an option for limiting the maximum size of the packages so as to optimise the time of data-sending and receiving.

The technology includes the DSTP (DataSocket Transfer Protocol) communication protocol used by LabVIEW, a protocol based on TCP/IP. It is possible to connect the DataSocket server using DSTP URL, as shown in the following example.

The URL below connects in the DataSocket server the data called Item1 which is being run in the same computer, the local computer, which may be the laptop where the SCADA_Laptop.vi. application is available.

Dstp://localhost/Item1

The URL below connects the data from the industrial PC to the tiem called Item1 in the DataSocket server which is running in the laptop connected to the wireless network.

Dstp://Direccion_PCPortatil/Item1

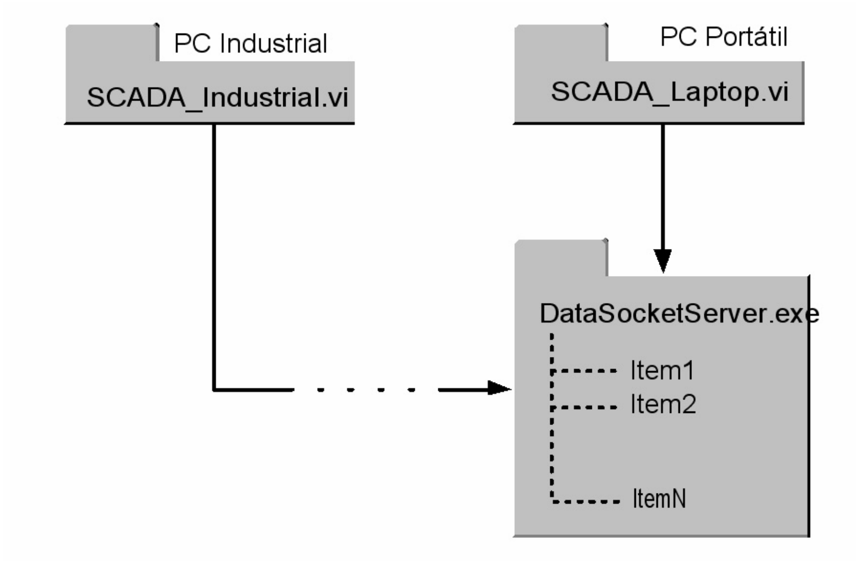


Fig. 6. DataSocket communication

In this way, data are exchanged through the wireless network between the industrial PC and the laptop with a single DataSocket server.

Once the above-mentioned applications are connected to the wireless network using DataSocket technology, the network can access from an external network, such as Internet, the SCADA_Laptop.vi application via the web server. For this, it is necessary to key in the corresponding address in a search engine such as Internet Explorer. The parameters of this URL specify on the one hand the web server address, which in this case is the ardes of the laptop, ando n the other hand the name of the corresponding application (SCADA_Laptop.vi). The full address is given below:

http://address.server.web/nameVI.htm

The SCADA_Laptop.vi application of the laptop and the application embedded on a web page which can be seen when the corresponding address is keyed in in Internet Explorer appear in Figure 7.

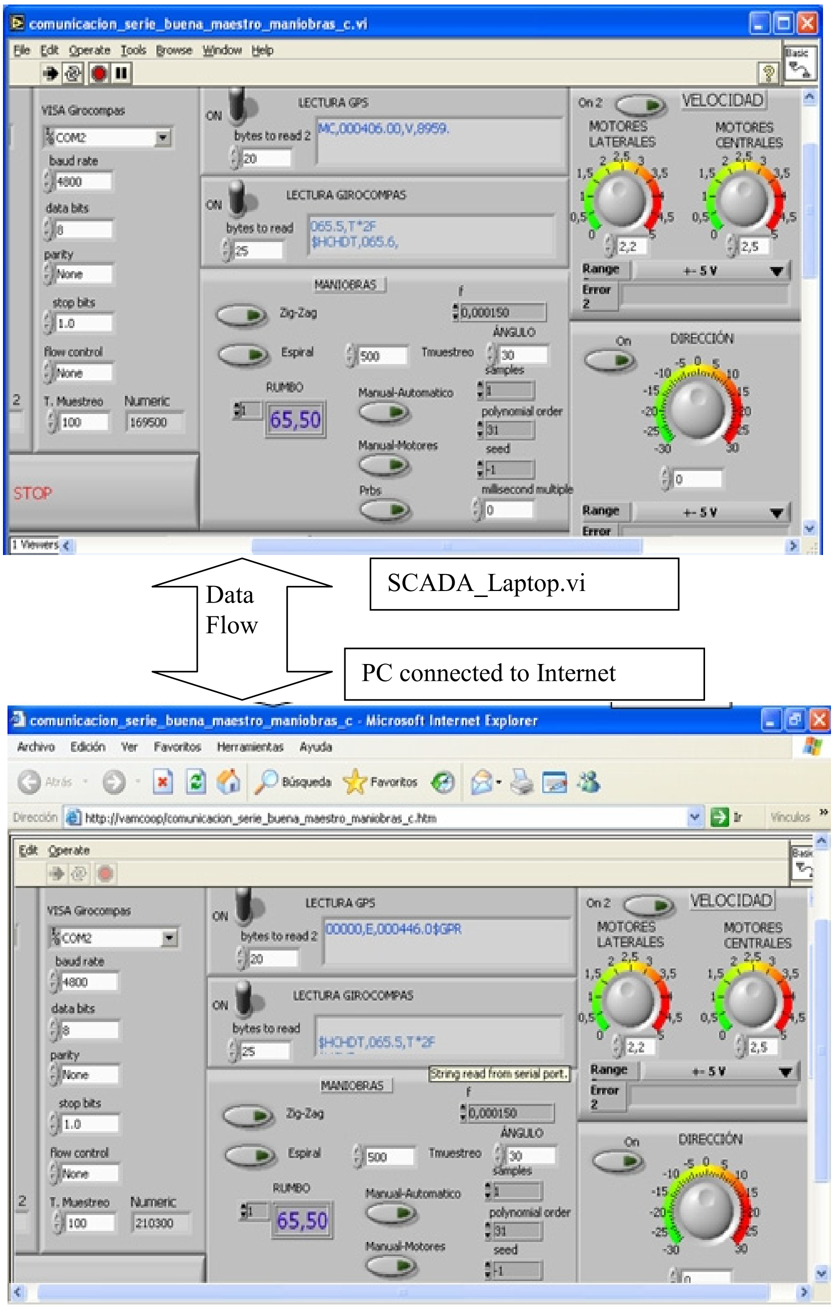


Fig. 7. Communication via web between the Industrial PC and the laptop

4 Sea Trials with the Test Platform

A ship may be capable of maintaining the projected path in the open sea, manoeuvring without danger in restricted ports and canals, and stopping within a reasonable distance, independently of the load and speed conditions and the external disturbances of current, tide and/or wind

Several tests are normally performed on a newly built ship, allowing the manoeuvring characteristics of the ship to be determined: evolution curve, zig-zag manoeuvre, "pull-out" manoeuvre and spiral manoeuvre (Lopez, 2004). These tests allow the characteristics of the dynamic behaviour of the ship to be measured, an indication of its stability in a straight-line trajectory to be obtained, the robustness and limitations of the control system to be assessed and the ship's behaviour in emergency situations to be evaluated.

Moreover, it is often necessary for the ship to undertake manoeuvres in cooperation with other ships, as in the case of aiding a damaged ship, towing manoeuvres and abarloamiento or mailing in proximity.

With this platform, the various sea trials required to determine the steering and manoeuvring characteristics of the autonomous in-scale model of the TF-120 ship or any other ship can be performed, as well as manoeuvres for actions in cooperation with other ships.

Moreover, the data obtained from the zig-zag tests and the evolution curve allow the identification (Amerongen, 1982) of a mathematical model of the ship which relates its gyro angle with the propulsion angle. With this mathematical model, controllers can be designed for the control of the above-mentioned manoeuvres and for the following of planned trajectories, all of which can be validated on the test platform as a stage prior to its actual implementation.

All of the elements that make up the system, as described in the above sections, are of the industrial type, such as the Industrial PC which is equipped to withstand the vibrations and movements typical of a marine vehicle.

4.1 Types of Sea Trials

To verify the manoeuvring characteristics of the ship, several sea trials can be performed which allow the ship's dynamic behaviour characteristics to be measured and its robustness and the limitations of the steering control system to be assessed (14^{TH} ITTC,1975). The manoeuvre characteristics can be obtained systematically maintaining or changing a preestablished heading and ship's speed.

To determine the efficiency of the ship's behaviour when maintaining the heading, the most suitable tests proposed are: direct and inverse spiral test and zig-zag manoeuvre with small rudder angles. To determine the quality of the behaviour in the change-of-heading manoeuvre, the recommended tests are the zig-zag manoeuvre, the 15º of rudder evolution test and the change-of-heading manoeuvre.

4.2 Development of Sea Trials with the Platform

The various manoeuvres described above can be performed in the experimentation platform developed.

In order to perform the change of heading required in each of the manoeuvres, the rotation angle of the in-scale TF-120 vessel model's turbojets is modified. Next, the data from the various instruments of the platform such as the electronic gyrocompass and the GPS are gathered and stored.

It should be noted that for the turning circle, the platform allows maximum rotation angles of around 30º to port and to starboard.

The zig-zag manoeuvre is achieved by modifying, from the remote laboratory, the rotation angle of the turbojet by 20º to starboard from the initial position of 0º (neutral position). Once this position is stored, the software support sends the command to change to a rotation angle of 20º to starboard, which is kept for a specified time and is then changed to 20º to port, and so on alternatively to starboard and port until a total of 5 changes in the turbojets are made.

In order to carry out the spiral manoeuvre trial in the remote laboratory, the marine vehicle must initially sail in a straight line. The rotation angle of the turbojets is then changed 25º to starboard and is kept thus until the system establishes itself. Then, the rotation angle is successively reduced by 5º each time until it reaches 25º to port. In the rotation angle range near 0º, the angle is decreased to values lower than 5º in order to obtain more accurate data.

The remote laboratory is capable of capturing the data on the heading of the vessel using a gyrocompass, of making a numerical derivation and thus obtaining the change of heading speed. This allows the pull-out manoeuvre to be performed. If the marine vehicle is stable for the straight line navigation, the change of heading speed, when the angle is modified from 20º to 0º, should drop by the same value to starboard as to port. If not, the vessel is instable.

The results obtained in the remote experimentation platform for the in-scale vehicle allow the behaviour of the real size vessel to be predicted. There is a formulation which relates the geometry, cinematics and dynamics of the in-scale model with the real vessel, so that a good approximation of what the behaviour of the real vessel would be like can be achieved .

4.3 Results of Sea Trials with the Plataform

Figures 8 and 9 below show the results of the evolution curve manoeuvre made with the platform.

Figure 8 show the curve for the evolution towards starboard. In the abscissa axis, the number of samples captured with a sampling period of 100 milisconds are represented. In the ordenates axis, the data on the heading measured with the electronic gyrocompass are shown as well as the gyro angle of the turbojets. In the case of the evolution towards port, a turbojet angle of 30º has been set. This figure shows a first phase of approximation, typical of the manoeuvre, in which the heading of the physical model remains constant with a turbojet angle of 0º. Then, the turbojet angle is modified to 30º, which is when the physical model begins to rotate towards port, and this turbojet angle is maintained until the model passes 360º twice to perform the full manoeuvre (Bech, 1968).

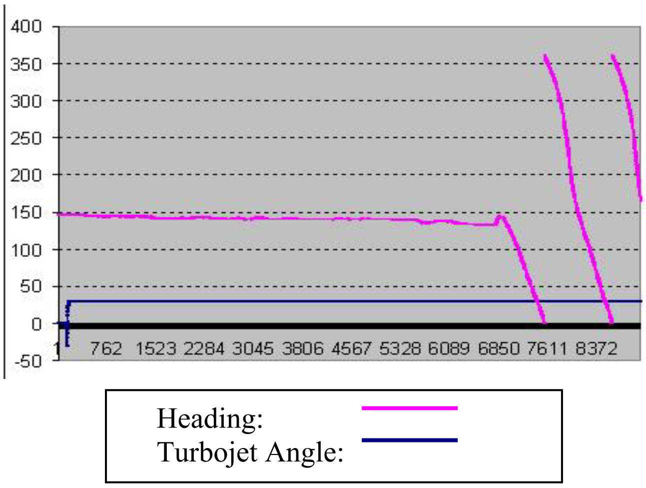


Fig. 8 Evolution towards port curve.

Figure 9 shows the evolution towards starboard curve, following the same philosophy as for the port manoeuvre. For this curve, a turbojet rotation of -30º has been set.

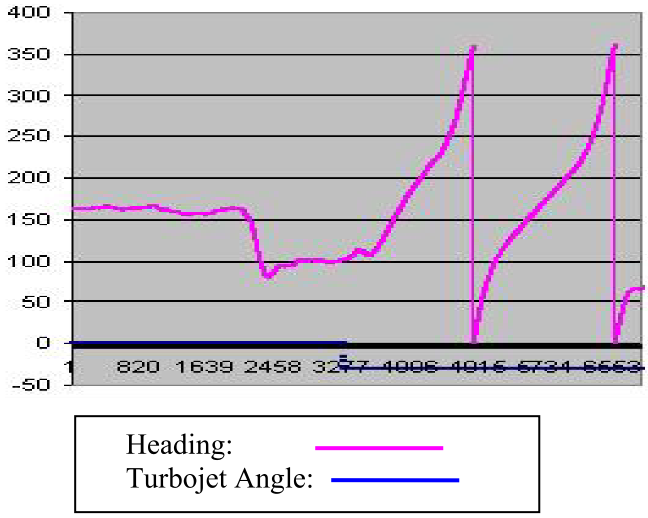


Fig. 9 Evolution towards starboard curve

In the development of these trials, a constant position of 0º degrees has been set for the bow flaps and 7,5º for the stern T-foil.

5 Conclusions

A test platform for marine vehicles has been designed in which an installation has been designed and set up with the actuators and instrumentation necessary for carrying out the data-gathering and control of the platform in such a way that an optimum monitoring of the trajectories can be performed. A software system has been designed for this platform which is equipped with a wireless network for communicating the vehicle with the laptop on land and there is also the possibility of accessing this network through Internet. The application designed deals with the gathering of data and the control of the physical model of the Turbo-Ferry TF-120.

In the development, construction and implementation of this platform, it was necessary to dedicate some considerable time to the adjustment and setting up stage, before achieving an optimum functioning of all the elements that make up the platform. It is essential, moreover, to carry out a thorough and exhaustive maintenance of the precision elements that make up the propulsion as these are affected by sea waters.

The correct functioning of the installation in the manual/automatic/manual operating modes has been verified. The correct functioning of the electronic gyrocompass has been checked, thus discarding any possibility of electromagnetic incompatability.

The distribution of the cabling of the electrical installation and the centralisation of all of the input and output signals in an electric box considerable reduces the time of detection of faults and maintenance. Moreover, the correct labelling of the cables allows the simple and fase modification and amplification of the installation. In this way, the platform is endowed with modularity for future modifications. The cabling and the magnetothermal protection of the motors has been adequately dimensioned to avoid faults in the worst posible cases.

It has been verified that the establishment of communications through the wireless network is correct. To do this, the communication between each PC and their access point and also the bidirectional communication between each PC and its point of access are verified. Once these chacks have been made, the data published on the web server of the Industrial PC can be accessed from the laptop without any problem. Communication between the 3G PCMCIA of the laptop and Internet is also verified as well as the correct communication between another or other PCs to Internet, whether it be using ethernet cable or other 3G PCMCIA cards. LabVIEW allows the connection to the web server of up to a maximum of five 5 PCs. The software support design is intended to make an efficient use of the CPU memory and resources.

This environment also includes stabilisation elements: a T-foil and the stern flaps which serve to stabilise the vertical movements of the physical model. Fins can be added without any great difficulty with their corresponding servomotor for the improvement of the sway movement of the physical model in the near future.

Acknowledgements

This work has been partially financed by the MCYT through project DPI2003-09745-C04-03.

References

14TH ITTC (1975) Discussions and Recommendations for a ITTC 1975 Maneuvering Trial Code. 14th International Towing Tank conference, Ottawa, September 1975, pp 348-365.

802.11g Wireless Access Point/Bridge WAP-4000 User Manual, (2004). Planet technology Corp.

Amerongen J. van (1982) Adaptative steering of Ships – a model reference approach to improved naoeuvering and economical course keeping Thesis (PhD). Delft University of Technology.

Bech, M.I. (1968). The reverse Spiral Test as Applied to Large Ships, Shipping World and Shipbuilder, pp. 1753-1754.

Bishop, Robert H.,(2004) Learning with LabVIEW 7 Express. Pearson prentice hall.

Fossen, Thor I., (1994) Guidance and Control of Ocean Vehicles. Marcombo

Hansen, D. A. (1996) Predictive Control and Identification: Applications to Steering Dynamics.

LabVIEW 7 Express User Manual (2003) National Instruments Corporation.

López E., Velasco F.J., Moyano E. and Rueda T.M. (2004), Full-Scale Manoeuvering Trials Simulation, Journal of Maritime Research, Vol.I., No.3, pp37-50.

NI DataSocket Server Help (2003) National Instruments Corportation.

Ollero Baturone, Aníbal, (2001)Robótica Manipuladores y Robots Móviles. John Wiley & sons.

Technical Manual KVH Digital Gyro Stabilized Sensor System.

TIM-LC, TIM-LF, TIM-LP System Integration Manual, (2004) u-blox AG.

CHAPTER 13

MILANA: a low cost glider used for building a map of the Barcelona sea bed.

J. AMAT[1], O. ESCOTE[2], M. FRIGOLA[2], X. GIRALT[2] and A. HERNANSANZ[2]
[1]Institut of Robotics, UPC-CSIC; [2] Research Group on Intelligent Robots and Systems, GRINS, UPC manel.frigola@upc.edu

A low cost versatile glider has been designed to navigate along the Barcelona shore and used to extract the profile of the marine platform. A virtual navigation system allows making a realistic reconstruction of the sea bottom from a set of real images obtained from a camera located in the glider.

1 Introduction

The main goal of the Milana project has been the development of a platform with auto control capacity, which allows obtaining high quality images of the sea bottom. The project has been motivated by the need of the Barcelona city Hall of supervising the evolution of the Barcelona shore sea bottom as a result of the control and filtering of residual waters in the city and surroundings (see Fig.1) and providing the means of showing the results to the society as well.

In order to obtain these high quality images, it is necessary that the distance between the camera and the sea bed changes only slightly along the navigation over the surface to be swept. The generation of a unique large size image, a mosaic, from a succession of images, requires that the resolution of the original images is kept as uniform as possible (see Fig. 2). The fitting of the set of partial images, obtained along the sweeping of the zone of interest, carries with it the adjustment of the zoom and the juxtaposition limits for every one of them.

Fig. 1 Map of the Barcelona shore with indication of the zone to be explored

Fig. 2 Acquisition of a sequence of images

The developed underwater vehicle (see Fig. 3), is not self-propelled, it is instead tugged by a support ship. Therefore the vehicle is simple and light compared to other explorer vehicles. The function of the umbilical cable is not only tugging the underwater vehicle, but also to transmit power and data from the glider.

To control the depth of the vehicle, the glider is provided with a pair of ailerons placed on its two lateral sides. The variation of the ailerons orientation is produced by the actuator located on the auxiliary boat that tugs the vehicle through the umbilical cable. The umbilical cable is composed of two traction wires and an additional one for power and communications. (see Fig.4).

2 Aileron Control

The depth control of the underwater vehicle is performed from the information obtained by the sensors available both in the vehicle and in the support boat.

Fig. 3 View of Milana, underwater vehicle for underwater exploration

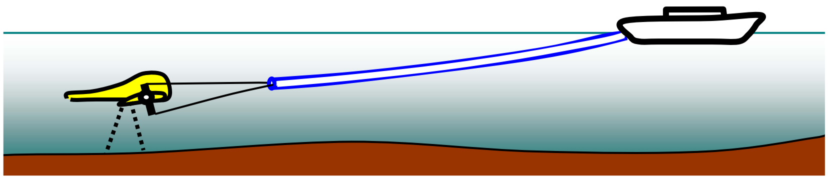


Fig. 4 Milana operating mode

.

The profile of the sea bottom is obtained in advance by the probe located in the tugging boat. The anticipation is related to the distance from the ship to the glider *X* that is first estimated from the measurement of length of the umbilical cable and from the glider depth *P*. The depth of the vehicle is measured using a pressure sensor and its attitude *Q* with a probe.

The control of the vehicle ailerons is carried out using a PD algorithm, (see Fig. 5). The derivative component of the PD algorithm is given by the depth variations, *H'*, obtained from the probe located in the boat, and applying a delay τ that is a function of the advancing speed and the horizontal distance, *X*, between the ship and the glider. The proportional component is given by the depth signal *P*, measured by the pressure sensor located in the underwater capsule.

The data obtained of the variation of distances from the vehicle to the sea bottom, Q, are used to compare the function (P+Q)(t) with H(t+τ). The comparison is used to update the value of the delay τ of the depth measured from the boat, as well as to correct the errors resulting from the evaluation of X.

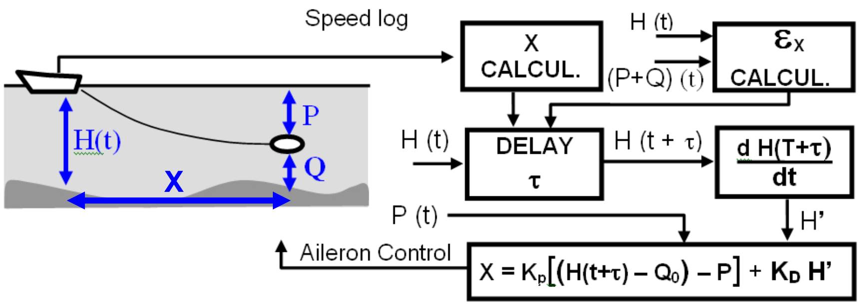


Fig. 5 Control of the ailerons of the capsule

3 Image Acquisition and Mapping

The objective of the image acquisition process is to capture and store all the required images to build the map in an efficient way. An efficient use of the storage capacity implies the possibility of mapping the images on-line. Unfortunately, this option is actually unachievable by regular computers due to the large areas that need to be covered and the big quantity of images involved. Relinquishing to an on-line mapping strategy to avoid the need of having available high performance computers on board, the whole process is divided into three main steps: *1)* sequential image acquisition and storage, *2)* off-line mapping and *3)* regular map mosaicing. The whole structure of the system is shown in figure 6.

3.1 Sequential Image Acquisition and Storage

There are two basic strategies that can be followed when acquiring images for mapping purposes: acquiring images at a constant rate, or acquiring them equally spaced. Acquiring images at a constant rate is the simplest option since only the sampling time needs to be adjusted. The storage unit does not need to be integrated into the vehicle control system, thus, cutting down the cost of the equipment. However, this strategy has a drawback if the vehicle speed is not constant. It either can produce too many images

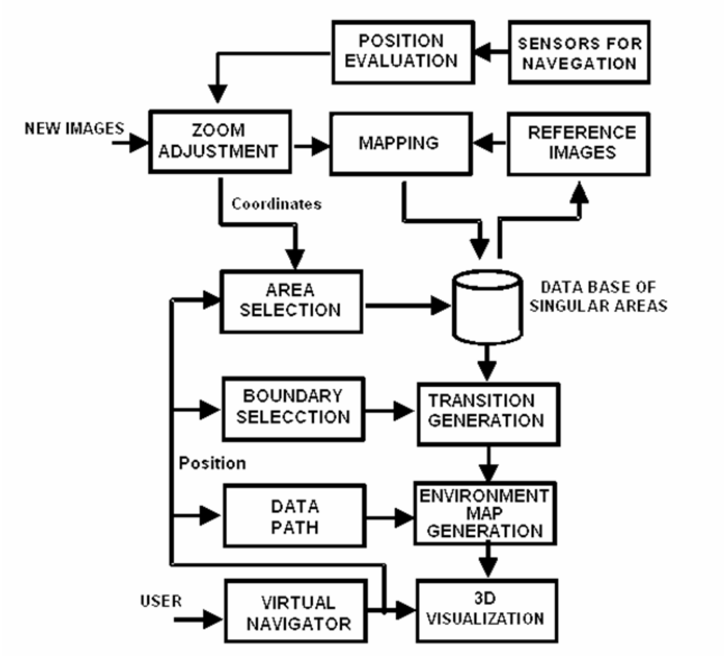


Fig 6. Overall system structure

when the ship slows down or leave areas uncovered when the vehicle speed increases. Therefore, the selected solution is the second one.

Acquiring equally spaced images requires taking a new image when the displacement surpasses a certain value. This value is usually a part (p) of the width (W) of the area covered by the last acquired image so as to assure an overlapping between two consecutive images. Considering that the cruise speed V_c of the ship and the altitude A of the glider have low dynamics compared to the image acquisition time, and being α the angle of the camera, the variable sampling period (T) that provides an approximate overlapping p between images is:

$$T = (1 - p) \cdot \frac{2 \cdot A \cdot \tan(\alpha / 2)}{V_c} \qquad (1)$$

The adjustment of the overlapping value p ($0 < p < 1$) results from a trade-off between incrementing the storage needs (high values of p) and the risk of producing "blind spots" (values of p near 0). Usually, the value p is set between 20 and 30% of the image size in order to guarantee a reasonable overlapping. Once the value of p is decided, a time-out event will trigger every snapshot after a lapse of T seconds. This period is regularly updated at each new instant using Equation (1) to adjust to the ship speed. Jointly with each image, the position (obtained from a GPS device), the ship heading and glider altitude are stored in a removable HD system. Every time a HD, is full the software for data acquisition changes to another removable device and notifies the user so as he can decide about the convenience of replacing it or not.

After sweeping the area with the camera glider (see Fig. 7) an off-line process will generate the map from the real images.

Fig 7. The Milana glider exploring the sea bottom

3.2 Off-line Mapping

The process of image mapping consists in finding the projection parameters of each image (Gracias 2000) (Negahdaripour 2001). Projection parameters are needed to render each image to its correct place. Therefore, the process of mapping relies on finding the image transformation (translations, rotations, zoom, etc) that best fits with its neighboring images. Here, translations are estimated directly from the data obtained from the GPS located in the ship, while the glider altitude and ship heading values are used for finding better image adjustments. The mapping procedure knows for each image the resolution (pixels/meter), data which is obtained from the glider altitude information and from the camera field of view, adjusting the image zoom accordingly. Also, the vertical rotation of each image is estimated using the ship heading. This correspondence (ship heading vs. glider orientation) is an approximation that is precise enough when the ship traces a straight line. In tacks this assumption is not correct, and furthermore, the relationship between the glider position, estimated from the length of the tugging cable, and the data from the GPS, does not allow estimating the real camera (glider) position. Nevertheless, if the maneuvers are correct the tacks will be outside the ROI (region of interest). Consequently, the mapping discards the images that are outside the ROI or those acquired when the ship is changing its course.

Nevertheless, even using the GPS and ship heading information, images do not fit adequately. The image positions need to be adjusted more precisely before the tessellation process. Tesselation is the most time consuming task because each image vertex has to be deformed in order to find the best adjustment with its neighboring images, however the image center is left quasi-unaltered. The reason for imposing this restriction is avoiding error propagation; otherwise each new adjustment would shift each image a little bit, producing at the end a deformed map. To avoid this distortion, only the vertexes position is altered a given percentage of the image size (see Fig. 8).

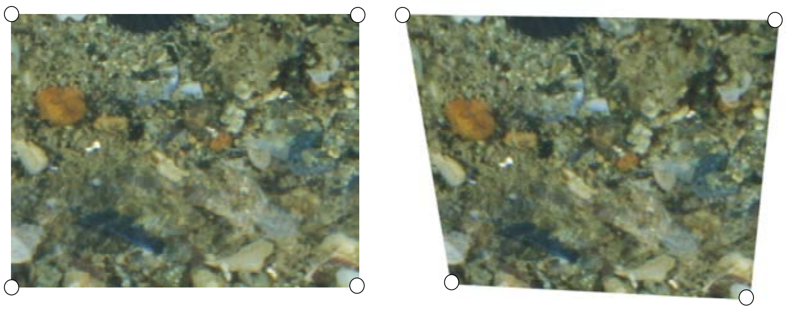


Fig. 8. Vertex deformation for fine image adjustment

The method proposed for measuring the correctness of an image adjustment is the joint entropy, as used frequently in image registration (Delso, 2004)(Pluim,2003)(Wein 2003).

$$H(A,B) = -\sum_{i,j} p(i,j)\cdot\log[p(i,j)] \qquad (2)$$

Where p (i, j.) stands for joint probability distribution, estimated from grey level image intensities. It is considered that two images fit correctly one to each other when one is transformed with respect to the other, in such a way that the joint entropy is minimized.

3.3 Regular Map Mosaicing

In the visualization phase the images that turn up into the screen of the browser are made by juxtaposition of orthogonal images. These orthogonal images are part of the grid created by the mosaicing process. This module is in charge of dividing the chosen area of navigation into sub-images (mosaic of orthogonal images) of 512 x 512 pixels. For obtaining a correct visualization in the unions between images, these sub-images are generated mixing the acquired images smoothly at the pixel level.

In order to fill in the empty zones of the resulting map it is necessary to generate synthetic transition images that give the impression of continuity between zones. The transition images are generated from the real images that characterize each zone. From the images that characterize each zone, new synthetic images are generated mixing their different textures according to two visual factors. The factors chosen for mixing textures are the distances between zones and binary masks that can provide a more "natural" appearance between zones. These binary masks, used as color keying between images, are created from the textures of the original images, making the transitions more abrupt when there are objects present in the scene. In order to not extend a transition indefinitely, the transition adjustments do not go beyond two meters of the border.

4 Virtual Navigation

With the goal to visualize the sea bottom using low cost equipment, a program that allows the virtual navigation in personal computers has been developed. This program uses the standard OpenGL libraries that offer reasonable performances when used with 3D graphic accelerator cards. The virtual navigator allows a smooth navigation while visualizing the mosaic of images of the sea bottom. Figure 9 shows the structure of the graphical interface. The program allows the visualization of the whole Barcelona sea bed in an interactive way: the user can interact with the browser choosing speed, direction, depth and angle of inclination of the virtual camera.

To be able to navigate continuously in real time the system limits the maximum distance of the scene that can be visualized. This limitation is due to the fact that visualization of extensive areas with a sensation of continuous movement in real time is not possible using conventional computers (Lindstrom 2002).

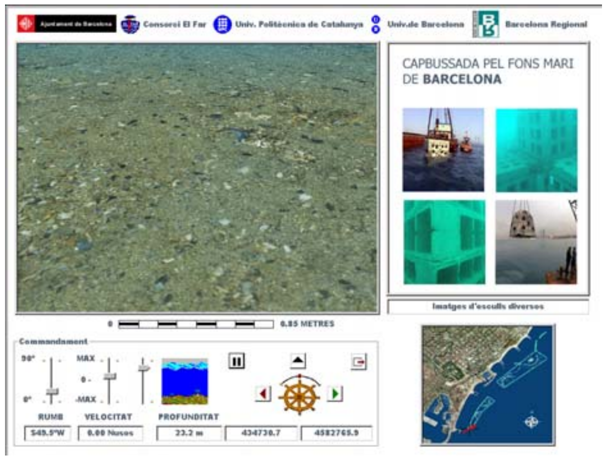


Fig. 9 View of the browser during virtual sailing along the sea bed.

During navigation the system reports extra information when the user approaches singulars structures. The system supplies information about type of artificial reefs that are being visualized, technical information (dimensions, weight, materials) and which biotope properties make it more suitable for accommodating one or another type of aquatic life (Yamamuro 2002).

5 Conclusions

The Milana robot has been tested in the process of obtaining images of the sea bottom in the Barcelona shore. The images have been taken at a distance of two meters of the sea bottom. The variations of the zoom produced by the fluctuations in height of the trajectory achieved by the robot are compensated through a rescaling of the size of each image. For that, a depth function, polynomial, obtained by extrapolation of the last four measures acquired by the probe endowed in the robot is used. The reduction of the fluctuations in height during navigation is achieved using the depth control function, which is obtained from the support ship. This depth function provides a set point some time in advance considering the variations of the orography of the marine platform. With this strategy, and tracking a bottom surface with soft gradients, the height oscillation band was less than +/- 40 cm for the 95% of the navigation time. These oscillations are considered short enough for acquiring snapshots that are later cut and resized so as to perform their fitting. These short limits have been achieved by reducing the surface of the ailerons that control depth. This reduction in size decreases the capacity of overcoming sudden reefs as well. At present, the capability of modifying the trajectory height is close to 20% of the vehicle speed, that is, advancing with a speed of 3knots (1.54 m/sec) the maximum depth variation is approximately 0.3 m/sec.

The robot Milana, the glider, can also be used in inspection tasks having linear fixtures such as cables, pipes or other similar elements.

Acknowledgements

Special thanks to the School of Nautical Studies of Barcelona, Group of Marine Geosciences of the Barcelona University, Barcelona Regional and Consortium EL FAR for their invaluable help.

References

G. Delso. Phd Thesis (in Spanish). Elastic registration of Multimodal Medical Images. Application to oncology, UPC Barcelona, 2004.

P. Lindstrom and V. Pascucci. Terrain simplification simplified: A general framework for view-dependent out-of-core visualization. IEEE Transaction on Visualization and Computer Graphics, 8(3):239–254, 2002.

J. P. W. Pluim, J. B. A. Maintz, and M. A. Viergever, Mutual information based registration of medical images: a survey, IEEE transactions on medical imaging, 2003.

Wolfgang Wein. PhD Thesis. Vision, Intensity Based Rigid 2D-3D Registration. Algorithms for Radiation Therapy. München, 2003.

N. Gracias and J. Santos-Victor, “Underwater video mosaics as visual navigation maps,” Computer Vision and Image Understanding, vol. 79, no. 1, pp. 66–91, July 2000.

S. Negahdaripour and P. Firoozfam, “Positioning and image mosaicing of long image sequences; Comparison of selected methods,” in Proc. of the IEEE Oceans 2001 Conference, Honolulu, Hawai, USA, November 2001.

Masumi Yamamuro et al. “Mapping tropical seagrass beds with an underwater ROV”. Proc. Recent Advances in Marine Science and Technology, 2002

CHAPTER 14

Towards automatized cooperation of ships in spill-over recovery scenarios

J. M. DE LA CRUZ[1], J. GIRON-SIERRA[1] , S. ESTEBAN[1] , J. RECAS[1] , J. F. JIMENEZ[1] , S. CIFUENTES[1] , C. LEON[1] , E. BESADA[1], J. A. LOPEZ-OROZCO[1], B. DE ANDRES[1] and J. FDEZ. PRISUELOS[1]

[1]Departamento de Arquitectura de Computadores y Automática, Facultad de Ciencias Físicas, Universidad Complutense de Madrid
jmcruz@fis.ucm.es

The initial step of a recently started research on marine crafts cooperation has been to decide, after some analysis, several marine operations requiring the cooperation of a set of ships, having virtual and/or physical links. The target of the research is to obtain procedures to be embodied into ships on-board computers to help captains to accomplish cooperative marine operations. This chapter proposes the case of two ships confining a spill over by towing a boom. The boom exerts an important interaction with ships heading. The case is modeled and simulated. Several phases of the operation are identified and, in each phase, the need of cooperative control is shown. The planning of the operation, together with trajectory optimization aspects are considered. Some initial experimental studies are also introduced.

1 Introduction

Recently we started a line of research on marine operations that require cooperation among several ships. The aim of the research is to propose ways of introducing automatization in these operations. In particular we are interested in protocols and cooperation modes. One of the ideas we are handling is to create a system of verbal orders to be understood both by captains and on-board computers, in order to have the ships participating in a certain operation being able to converse.

The general procedure of the research is to propose certain archetypical scenarios, analyze them in simulation and, if possible, conduct scaled down experiments. For instance, we consider scenarios with virtual or real links between ships, like in cases of formations, towing, transfers, etc. There are other scenarios with no links, but having positional restrictions, which may be caused by real obstacles or by the need to avoid certain zones (for instance, due to traffic rules). Rescue or patrolling scenarios may require scanning methods. Etc.

Our general purpose is to study in the marine context the topic of cooperation between mobile physical agents. An important peculiarity of marine surface vehicles is that sway motion cannot be directly controlled, unless dynamic positioning actuators are provided. In our research on marine cooperation we take advantage of the experience acquired with seakeeping control (Giron-Sierra, Esteban, et al., 2005), since for this research we developed and built autonomous scaled ships for experimental purposes. A relevant reference about ship control is (Fossen, 2002).

The topic of multi robot cooperation (Cao, et al., 1997; Parker, 2000; Farinelli, et al., 2004) is related with the broader context of multi agent interaction (Billard, 2004; Liu, 2001; Weiss, 1999). Interesting new aspects have been introduced in (Kumar, et al., 2005). In particular, formation control (Balch and Arkin, 1998; Tanner, et al., 2004) has been developed to a large extent. Formation control has been translated to the marine context by (Ikeda, et al., 2005), on UAVs, and by (Skjetne, et al., 2003).on ships.

Let us cite several papers on marine crafts coordination: (Johansen, et al., 2003) on towing, (Kyrkjebo and Pettersen, 2003; Kyrkjebo, et al., 2004) on ship rendezvous operations, (Morishita, et al., 2004) on offloading operations, and (Soetanto, et al., 2003) on coordinated motion control.

In this paper we deal in particular with two ships towing a boom for spillage over confinement. It is a simplified scenario that can be expanded to more linked ships. The case has been already introduced in some of our recent papers (Giron-Sierra, Jimenez, et al., 2005), but now we want to work it more in detail and show some preliminary experimental results.

The order of the chapter is the following: first we describe the spillage scenario to be considered, then we obtain a mathematical model of the dynamics and conduct some experiments in simulation, these experiments show the control problems that should be solved in a cooperative manner, then we attack two interrelated problems: the trajectory optimization and the planning of team maneuvers, then we use scaled down autonomous ships for experimental checking. The chapter ends with some conclusions and a look at future research.

2 The Spill Over Confinement Scenario

Figure 1 shows a conceptual view of the scenario. Two ships depart from the coast, towing a boom, and must obtain a good confinement of a spillage.

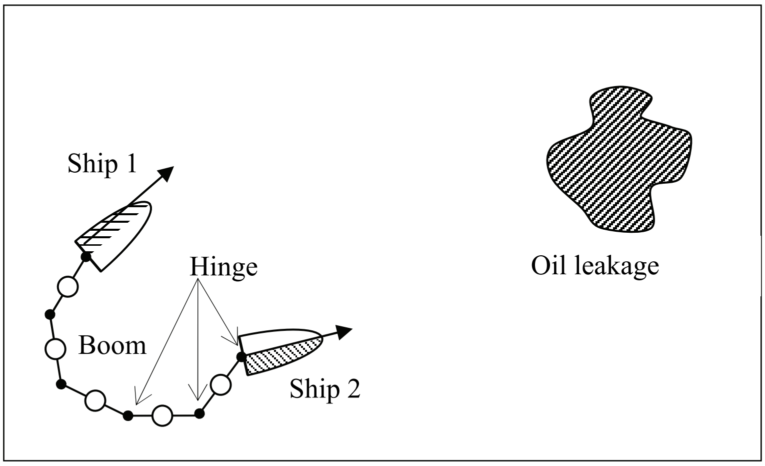


Fig. 1. Conceptual scenario

We can distinguish four main steps in the operation:

- Start towing the boom
- Some maneuvers for approaching the spill over
- Deployment
- Closing and towing

There are several problems to be solved in each step, for instance, the optimization of the approach angle to the spillage (which probably will have an irregular shape). The team organization could vary for each step. For instance, ship 1 position could be appropriate for the ship 1 to be the chief in certain circumstances, but this can change in other circumstances so it could be more convenient the ship2 to be the chief. Or even none of them should be the chief. There are also control problems, since the two ships physically interact.

3 Model of the Two Ships with the Boom

We suppose the boom can exert a significant torque on the ships. That implies an interesting course control problem. Figure 2 shows the variables considered to establish a mathematical model of the dynamics of the two ships towing the boom:

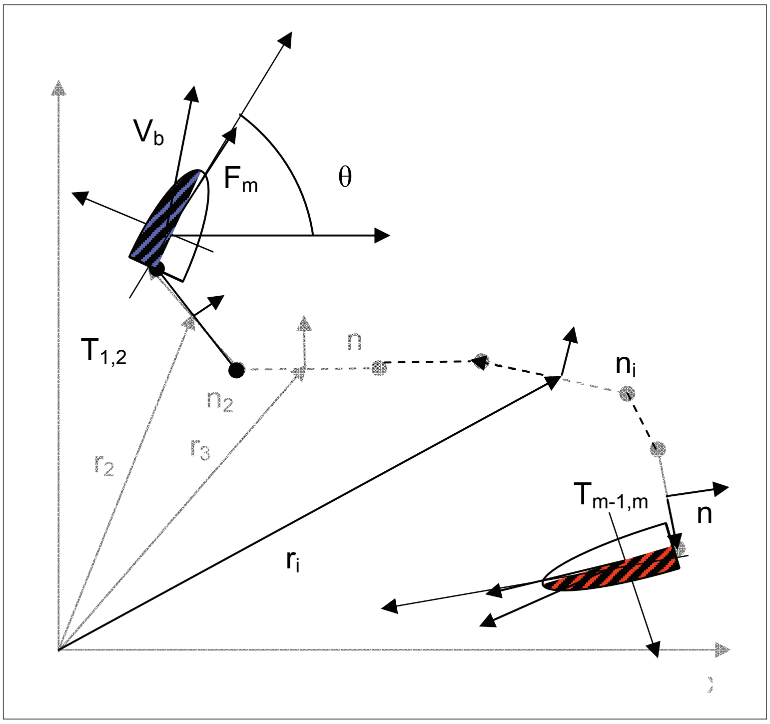


Fig. 2. Variables of the mathematical model

The motion of a ship, in the x-y plane, can be modeled as a first simple approximation with the following equations:

$$m_b a_{bx} = [F_m - \mu_l (v_{by} sen(\theta) + v_{bx} \cos(\theta))]\cos(\theta) \\ - \mu_t l_s (v_{bx} sen(\theta) - v_{by} \cos(\theta)) sen(\theta) \tag{1}$$

$$m_b a_{by} = [F_m - \mu_l (v_{by} sen(\theta) + v_{bx} \cos(\theta))] sen(\theta) \\ - \mu_t l_s (- v_{bx} sen(\theta) + v_{by} \cos(\theta))\cos(\theta) \tag{2}$$

$$M - \mu_a l_s \omega_b = I_b \alpha_b \tag{3}$$

The motion of the boom, which is attached to the ships, is deduced by first considering a link and then combining several links (imposing the corresponding closing conditions). The motion of a single generic link is given by:

$$\vec{T}_{i,i+1} - \vec{T}_{i-1,i} - \frac{\left(|\vec{v}_i \cdot \vec{n}_i| s + |\vec{v}_i \cdot (-\vec{p}_i)| q\right)}{|\vec{v}_i|} \vec{v}_i = m\vec{a}_i \tag{4}$$

Were, T_i represents the strain in a tip of the link, v_i is a normal vector, p_i is a vector opposite to the sense of the link, q and s represent longitudinal and perpendicular drag coefficients and m is the mass of the link.

And the link rotation is described as:

$$(\vec{T}_{i,i+1} \cdot \vec{n}_i) l + (\vec{T}_{i-1,i} \cdot \vec{n}_i) l - A\omega_i = I\alpha_i \tag{5}$$

Were l is half the length of the link, A is a drag coefficient, and I is the moment of inertia.

The closing condition is,

$$\vec{r}_i - l\vec{p}_i - l\vec{p}_{i+1} - \vec{r}_{i+1} = 0 \tag{6}$$

4 Simulation of the Team Motion

A simulation environment has been developed in C++, with the ships being objects with dynamic behavior (actually, the ship objects have autonomous decision capabilities, so they can be considered as agents).

Let us consider the first step of the operation. Both ships start with a stretched boom. So much stretching is not very realistic, but it helps to see initial problems. Figure 3 shows a screen of the simulation.

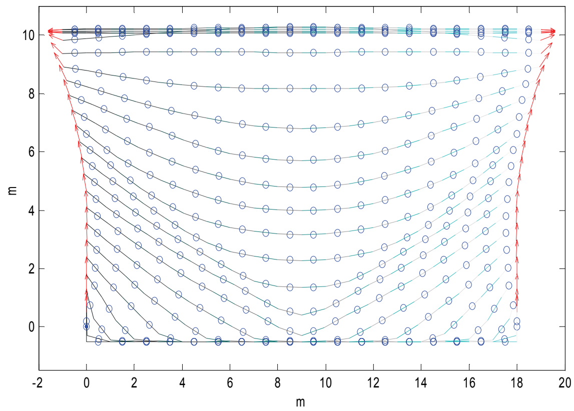


Fig. 3. Simulation of the first operation step

Both ships try to go in parallel, along a straight path. The results are computed for a boom composed of twenty elements, when no correction of course is applied to the ships. The arrows in the extremes represent the successive positions and orientations of both ships. The boom has been represented by lines with a circle in the centre of each element. The boom pulls the stern of the ships, and the ships rotate. Eventually, both ships make a tug of war. The need of a control action on the rudders, to counteract the boom tug, is clear.

Let us consider now a turn. Figure 4 shows a screen of the simulation.

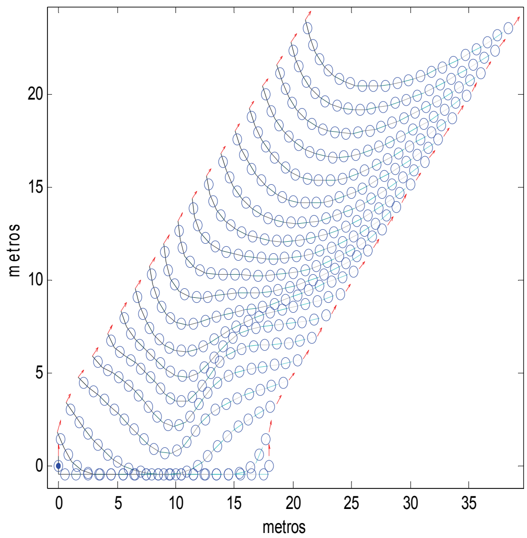


Fig. 4. Simulation of a turn

The turns have been performed controlling individually the course of the ships. As a result, when the ships reach their course set points, the ship located at the left end of the boom has fallen behind.

It is clear that such maneuver requires some kind of coordination to be correctly performed.

These results are clear examples of how it is necessary to employ a common control strategy for both ships under this particular scenario.

The basic control system has been implemented taking into account that both ships should perform encompassed maneuvers. As a consequence, the set points are established in two levels:

- A first level allows dealing with general requirements that should be negotiated at a higher level; (examples are a common cruise speed or a common course). These set points will depend on the particular operation stage but they have coarse influence in the propulsion and rudder systems of the ships.
- A second level involves a finer outlook on the system state, establishing set points for shared variables, such as mutual distance or common alignment. In this way, a subtle but necessary distinction has been performed between common variables: those variables which should have the same value for both ships, and shared variables: those variables which value depends on both ships at the same time.

Coming back to the particular scenario under study, speed control for each ship depends on common speed set point but also in the relative orientation of its course with the line that links both ships. Figure 5 shows a schematic view of the effect of this speed control.

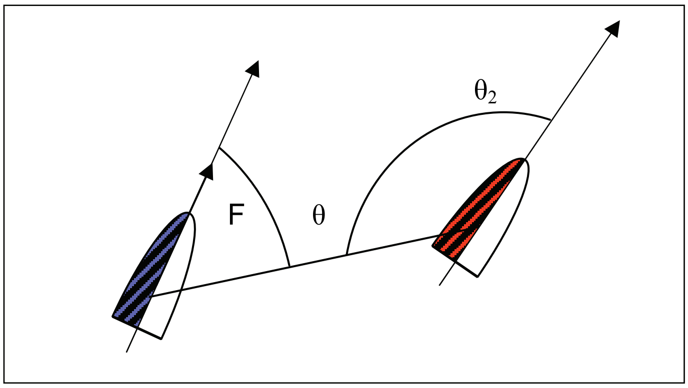


Fig. 5. Alignment control: speed control is affected by the value of θ_i

The set point for θi is $\pi/2$, a larger angle slows down the ship and a smaller one speeds it up.

On the other hand, course control, depends on the distance between the ships, There is a set point for this distance and ships will turn inwards or outwards their common trajectory accordingly.

This two shared variables plus a simple PID feedback control system appears to be enough to control the system properly.

Figure shows 6 a more complex maneuver, in which the ships, departing from repose, take a 45º course and at the same time approach each other till reaching a predetermined 'cruise distance' between them.

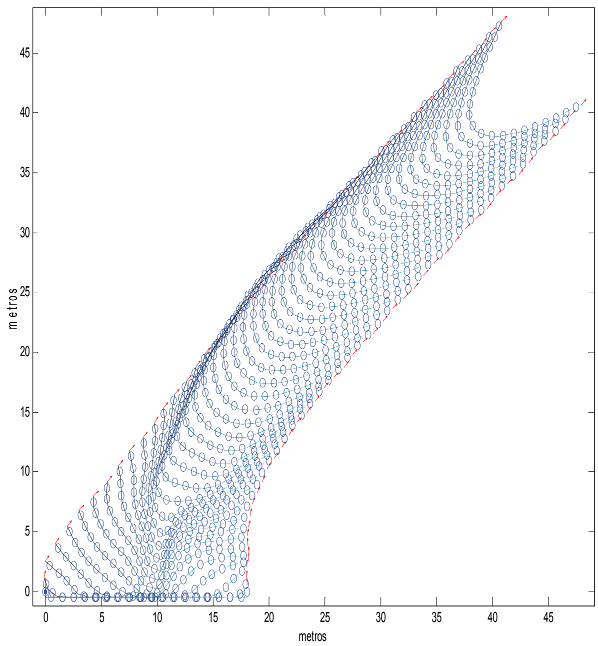


Fig. 6. Combined maneuver, including a turn and the mutual approach of the ships.

As can be seen in figure 6, both ships take gradually their new course, covering the left ship a large distance.

At the end of the maneuver, the ships are correctly aligned with the normal direction to their common course. It is also valuable to pay attention to the movement of the boom during the maneuver. As can be seen, the boom has its own dynamics induced by the ships motion. The left ship speed up to overcome its larger trajectory, giving the boom a tug. This jerk propagates along the boom shifting it towards the left ship. This effect reaches a maximum shortly after the ships have reached their course set point. Eventually, the boom stabilizes again, taking a typical catenary's shape.

This introduces another set of restrictions: speed and course changes must take into account their effect over the boom dynamics, otherwise the strains in the boom could damage it or seriously disturb the system performance.

The problem can be tackled in several ways. One possible approach is to employ an upper supervisory level which, according to the current state of the system, performs a path planning to achieve a safe trajectory until the desired set point is eventually reached.

Figure 7 shows an example of a course change of 135º. In this case, the path planning has changed the course set point for the ships gradually to assure a minimum jerk on the boom. As a consequence, the turn takes a radio of roughly 100 meters but the boom experiment almost no lateral shift.

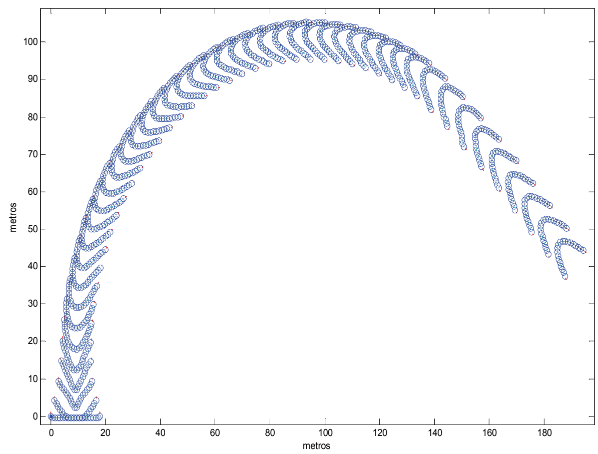


Fig. 7. A smooth 135º turn.

5 Multi Pseudo Bang Bang Control Optimization for Ship Trajectory Planning

Usually the ships participating in a scenario must go from one location to another, and must change their attitude. We want to calculate before the operation satisfactory trajectories for the ships.

The main idea in this section is to mimic the typical maneuvers with cars, which usually are of the type “partial” bang-bang (turn the wheel first to one side, then to the other side). When we say “partial” we refer that frequently it is not necessary to reach the wheel angular limits. The trajectory obtained is of sigmoid nature. The reason for this type of maneuvers is that cars are non-holonomic.

Conventional ships are usually non-holonomic. Moreover, contrary to the normal behavior of cars, the ships exhibit sway motions when turning.

There is an increasing interest on the control of non-holonomic vehicles, moreover when coordination of multiple vehicles is desired (Kumar, et al., 2005; Bloch, 2003; Baillieul and Suri, 2003).

We are going to consider three phases in a trajectory, say from haven to a ship in danger near the coast. The first phase is devoted to abandon the haven and get a good heading towards the target. The last phase is devoted to take the right attitude to get close to the target. The middle phase just connects the other phases. We shall consider a partial bang-bang for the first phase and another partial bang-bang for the last phase. The middle phase may be an arc.

It turns out that planning is a complete topic, with old roots in the robotics context. An extensive review is given by (LaValle, 2006). There is a distinction between path planning, which is geometry, and trajectory planning that also considers time. A closely related topic is trajectory optimization, which is reviewed in (Betts, 1998).

In this section, the trajectory is embedded into the problem of control actions along time. The results we are to obtain are directly applicable by ships, since what is obtained is a sequence of control actions that the ships can implement (the control authority is inside the range allowed by the ship actuators).

The problem considered in the section involves a multiobjective optimization (Miettinen, 1999). A good approach for this kind of problems is to consider Pareto fronts, trying to get appropriate agreements. This kind of criteria can be easily considered in the general context of Genetic Algorithms (Goldberg, 1989; Michalewicz, 1999).

Taking advantage of our previous experience with genetic algorithms (Esteban, et al., 2002), we state the partial bang bang maneuvers in terms of chromosomes. A multi-objective criterion to be optimized by the ship trajectory is established. The genetic algorithm gets with modes computational cost a satisfactory solution. This solution is a simple sequence of 5 control orders (each one consists in a time and a torque constant value to be applied from this time to the next).

The control planning method can first be applied off-line, to obtain a control plan for the experimental ships, and then can be re-applied on-line to compensate for perturbations effects. This seems to be an interesting approach in light of the Brockett theorem (Brockett, 1990; Aguiar and Pascoal, 2002).

In relation with the use of Genetic Algorithms for marine path planning, there are some papers like (Kwiesielewicz, 2000; Tan, et al., 2004; Alvarez, et al., 2004) for AUVs

5.1 Mathematical Model of Single Ship Motion

The mathematical model we take for the ship is a subset of the standard six differential equations (Lloyd, 1998), three for longitudinal motions and three for lateral motions, that constitute a linear approximation with variable coefficients (added masses, damping, restoring forces).

The equations selected for the research are the following:

$$
\begin{aligned}
&(m+a_{11})\,\ddot{x}_1 + b_{11}\,\dot{x}_1 = F_1 \\
&(m+a_{22})\,\ddot{x}_2 + b_{22}\,\dot{x}_2 + a_{26}\,\ddot{x}_6 + b_{26}\,\dot{x}_6 = F_2 \\
&a_{62}\,\ddot{x}_2 + b_{62}\,\dot{x}_2 + (I_{66} + a_{66})\,\ddot{x}_6 + b_{66}\,\dot{x}_6 = F_6
\end{aligned}
\tag{7}
$$

where suffix 1 is surge, suffix 2 is sway, and suffix 6 is yaw. Note that there are no restoring forces.

Using Laplace transform to obtain polynomials, and then using Cramer's rule with the following terms:

$$
\begin{aligned}
&N = [(m+a_{22})(I_{66}+a_{66}) - a_{26}\,a_{62}] \\
&F = [b_{22}(I_{66}+a_{66}) - a_{26}\,b_{62}] \\
&G = [b_{26}(I_{66}+a_{66}) - a_{26}\,b_{66}] \\
&H = [b_{66}(m+a_{22}) - a_{62}\,b_{26}] \\
&J = [b_{62}(m+a_{22}) - a_{62}\,b_{22}]
\end{aligned}
\tag{8}
$$

We finally obtain

$$\begin{pmatrix} \ddot{x}_2 \\ \ddot{x}_6 \end{pmatrix} = \frac{-1}{N} \left\{ \begin{pmatrix} F & G \\ J & H \end{pmatrix} \begin{pmatrix} \dot{x}_2 \\ \dot{x}_6 \end{pmatrix} + \right.$$
$$\left. + \begin{pmatrix} (I_{66} + a_{66}) & -a_{26}\,(I_{66} + a_{66}) \\ -a_{62}\,(m + a_{22}) & (m + a_{22}) \end{pmatrix} \begin{pmatrix} F_2 \\ F_6 \end{pmatrix} \right\} \tag{9}$$

The coefficients in these equations have been determined with experiments using the scaled ships in the basin.

Figure 8 shows an example of ship trajectory, obtained for the maximum yaw torques that the ship actuators (we use waterjets) can yield, except for the middle arc where a soft yaw torque has been applied. The dimensions are given in meters; the experiments must fit the dimensions of the basin.

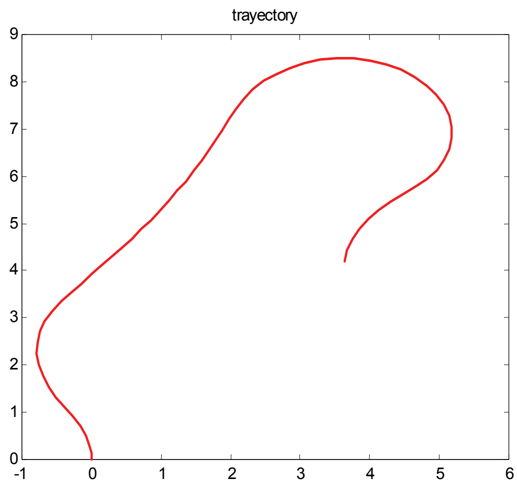


Fig.8. Example of ship trajectory using five arcs.

5.2 Genetic Procedure

The two main issues for the application of Genetic Algorithms are the codification in terms of chromosomes and the definition of a fitting function.

For the chromosomes we use integer quantities, from 0 to 999. Suppose for instance that we have the number 374, corresponding to the yaw torque in the middle arc. That means that 37.4 percent of the maximum yaw torque will be applied in this arc. The chromosomes contain ten integer quantities, corresponding to start time and yaw torque for each of the five arcs. Figure 9 shows a chromosome. Every cell in the diagram has three integer numbers inside.

T1	YM1	T2	YM2	T3	YM3	T4	YM4	T5	YM5

Fig.9. Structure of chromosomes.

Each chromosome means a specification of the control effort (partial bang bangs) along the operation. By using the ship mathematical model, the GA evaluates the chromosomes, obtaining and evaluating the ship trajectory obtained by the control efforts. This is the GA fitting function. A multi-objective criterion is defined for trajectories evaluation. There are two groups of objectives:

- The primary objectives are to get precisely to the target position, with the specified attitude angle.
- The secondary objectives are time and energy minimization.

Both groups are treated with Pareto fronts. When a good compromise is obtained in the first group, the Genetic Algorithm searches for another good compromise in the second group and then, it comes back to refine the first compromise, and so on, till no significant improvements are gained keeping iterating.

5.3 Some Results

Figures 10,11 and 12 show the results obtained for a ship starting from position (0,0) and attitude pi/2, and going to position (20,60) and attitude –pi/2. The curves represent the trajectory and the partial bang bang control efforts.

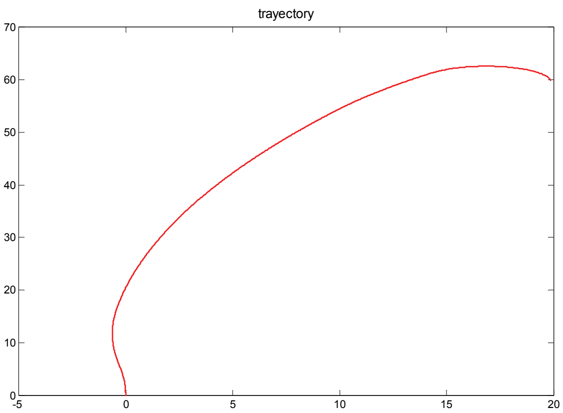


Fig. 10. Example of five arcs trajectory and partial bang bang control

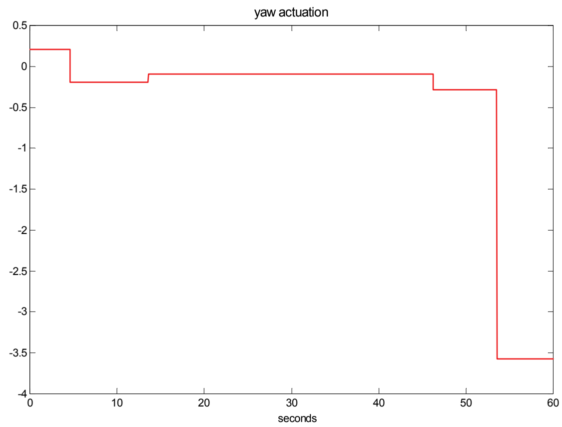


Fig. 11. Partial bang bang control for the example above.

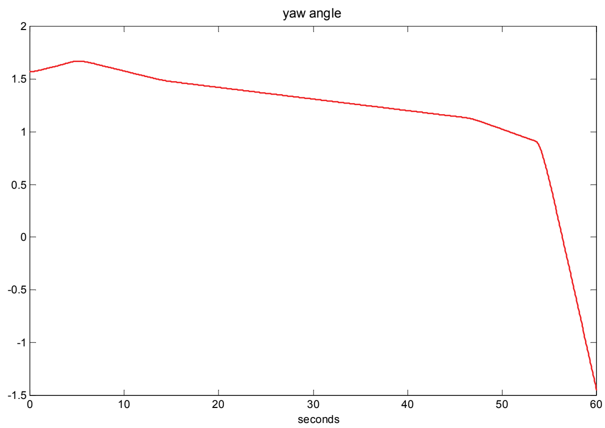


Fig. 12. Evolution of ship's yaw angle for the example above.

5.4 Two Ships

The control planning method just introduced was conceived taking into account ship formations, where the motion problem may be decomposed into two: to move the "centre of gravity (CoG)" of the formation along a feasible trajectory, and to keep or change the formation along the trajectory (with reference to the CoG). However, we also recently considered the control planning for several ships at the same time. We started with two ships, extending in a simple way the partial bang bang idea.

The scenario consists in two ships that must go to a target position and attitude angle, keeping their mutual distance and angle into a certain range (for instance, it may happen that both ships are towing a boom).

Since in a turn one of the ships must go faster than the other, surge force control was introduced. To restrict the heuristic searching space, one of the ships has much less surge force range than the other: that means that one of the ships will advance at almost constant speed, while the other can change it in a more flexible way).

The chromosomes were extended to include five arcs for ship1, and another five arcs for ship2. In each of the arcs the chromosome specifies its timing, surge force and yaw moment.

A "virtual ship" was introduced, that goes in parallel to ship1 at fixed distance D. The position error of the ship2 with respect to the virtual ship is measured along each candidate trajectory. The maximum positional error is minimized as one of the objectives to be taken into account by the multiobjective optimization.

Figures 13, 14, 15 and 16 show several aspects of the solution obtained for the case of coming from the haven with pi/2 to a target with –pi/2.

Figure 13 shows with dashes the "virtual ship" trajectory, ship2 stays between the virtual ship and ship1.

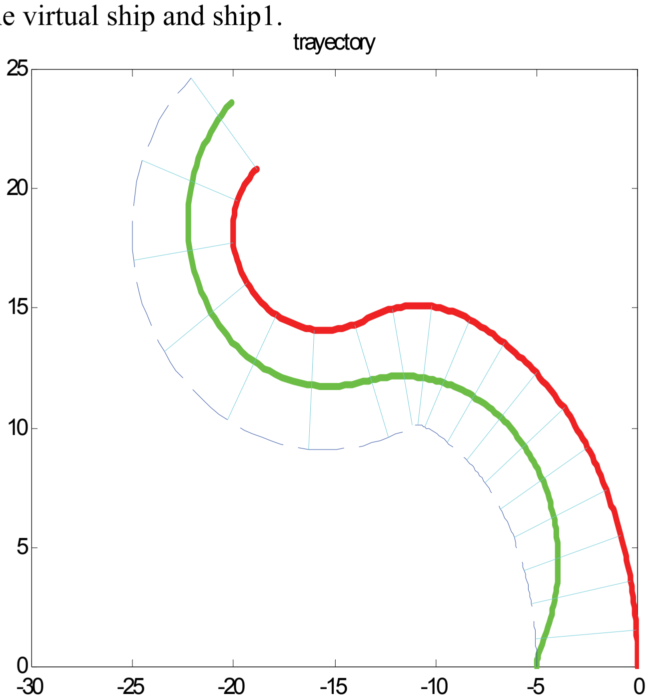


Fig. 13. Trajectories of ship1, virtual ship, and ship2 with partial bang bang control.

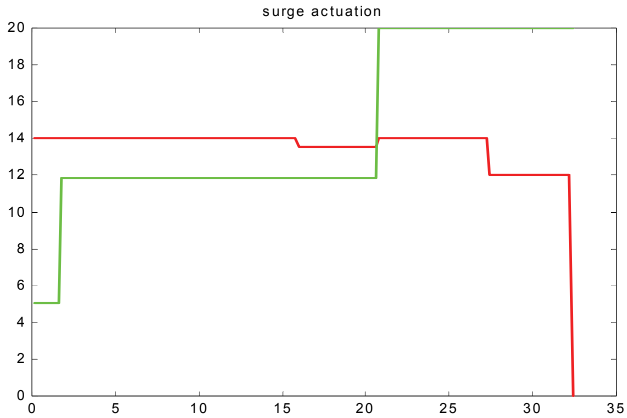


Fig. 14. Surge controls for the two ships.

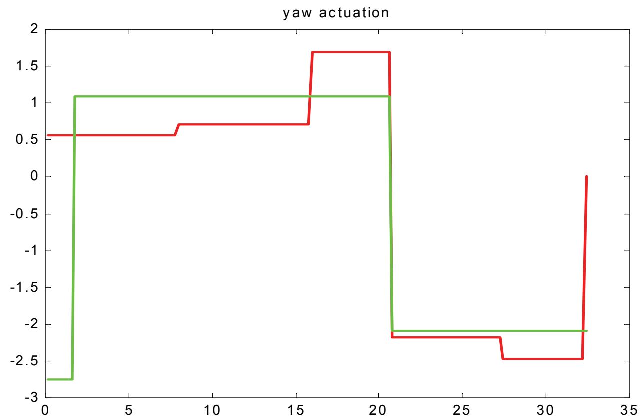


Fig. 15. Yaw control for the two ships..

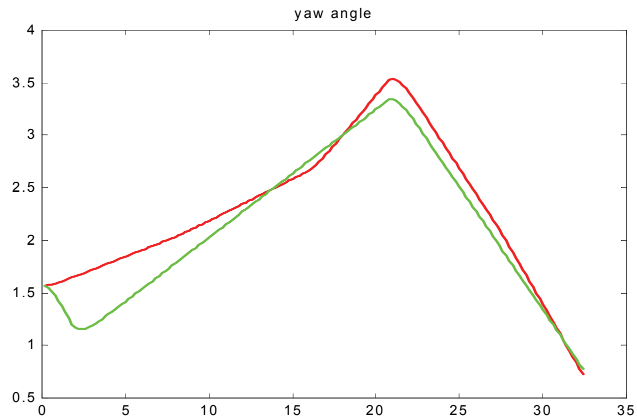


Fig. 16. Yaw angles of both ships along their trajectories.

The control profile is described with simple vectors (with ten float numbers) that can be easily injected into ship on-board processors for experimental execution.

The partial bang bang idea can be extended to in-line application, during actual ship motion. For instance, a mathematical model of the ship can be used on-board to determine when the real trajectory deviates too much from the desired trajectory, and then intervene with a partial bang bang corrective maneuver.

6 Autonomous scaled ship for experimental studies

As said before, in our research a set of scenarios requiring ship cooperation have been selected. For instance spill over confinement, different types of formations, and rescue operations. The target of the research is to propose ways of automatization, by means of verbal orders between computers on the bridge of the ships involved in an operation (under the supervision of captains).

Indeed the research can progress only in terms of computer simulations. However, we wish to have experimental verification, since in a research with a marked accent of initial exploration it is fruitful to run experiments, to promote discovering.

Since some experience has been gained with our research with scaled fast ships, for seakeeping control purposes, we take advantage of it to develop a set of autonomous scaled ships for robotic cooperation studies on water surface.

In this chapter we are considering an experimental scenario where two robot ships tow a boom, and deploy it for spill over confinement. The autonomous ships must know where they are, must control their motion, and must exchange messages. These are the main functions that an on-board system must assume.

Fig. 17 shows a block diagram with the hardware that executes the functions that have been implemented for our ships. This is the hardware that we are going to control.

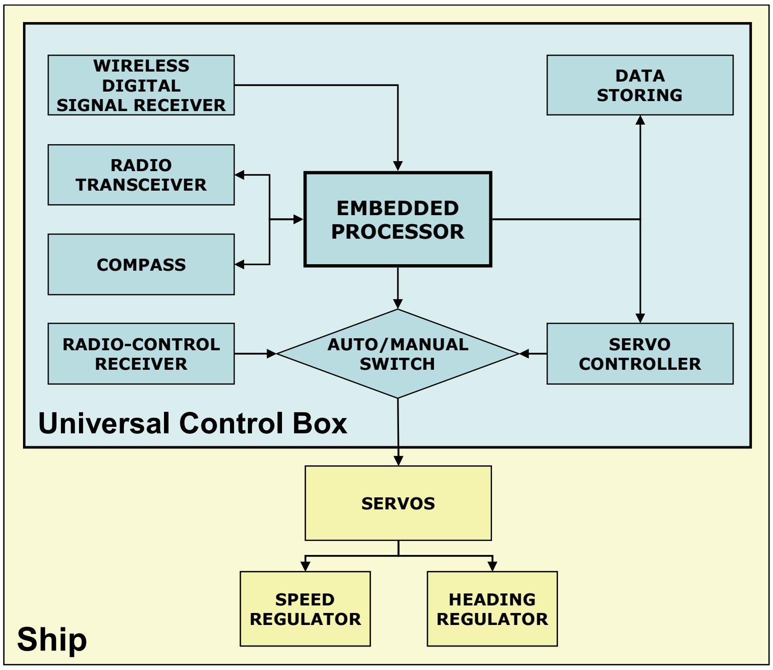


Fig. 17. Block diagram of the ship on-board system

The heart of the on-board system is a low-power small size embedded PC. We added a digital compass with good precision and adequate sampling rate. A digital packet radio link has been included, for the ships to exchange messages. This radio manages important things like the media access control. With the servo controller, the system can easily control many servos, 8 in our ships: this has been useful for ship's heading and speed control. We also have included a digital wireless receiver, which we use to give simple digital signals, like reset and some others that the processor can handle. The rest of the system is a manual radio-controlled system, used to move and rescue the ships in case of fail. The system then connects to some servos, the actuators that will move the ships.

The on-board system has been put into a hermetic box, with connectors. This box can be seen as a black box; only taking care of its inputs, power and control signals; and outputs, servo connections and info signals (status LEDs, in this implementation). This way the box can be used in a wide range of autonomous vehicles, even real scale, for many different applications: not only ships, although it is heavily focused on this. This is the reason why we denote this as the "Universal Box". Fig. 18 shows a photograph of it.

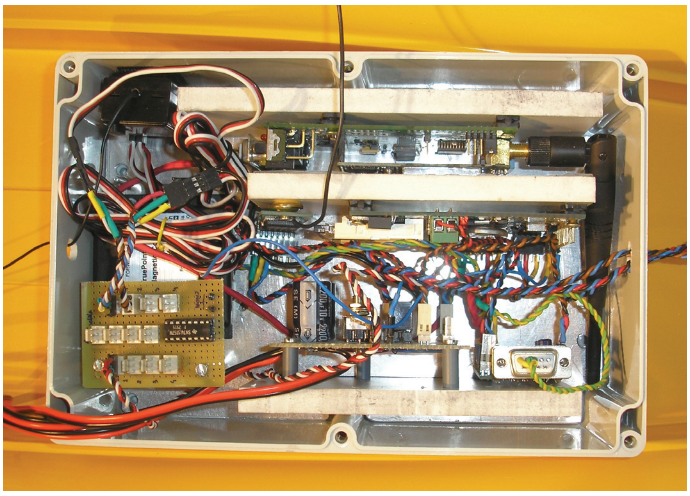


Fig. 18. The 'Universal Box'

For our initial experiments two scaled ships have been completely built, and three more are under way. The ships use orientable waterjets for propulsion and motion control. One box is inside every ship. Fig. 19 and Fig. 20 show a photograph of them.

Fig. 19. Frontal view of the scaled ships

Fig. 20. Lateral view of the scaled ships

7 A ship Language for Cooperation

Fig. 21 shows a diagram with the software architecture of the program, which governs the ship behavior:

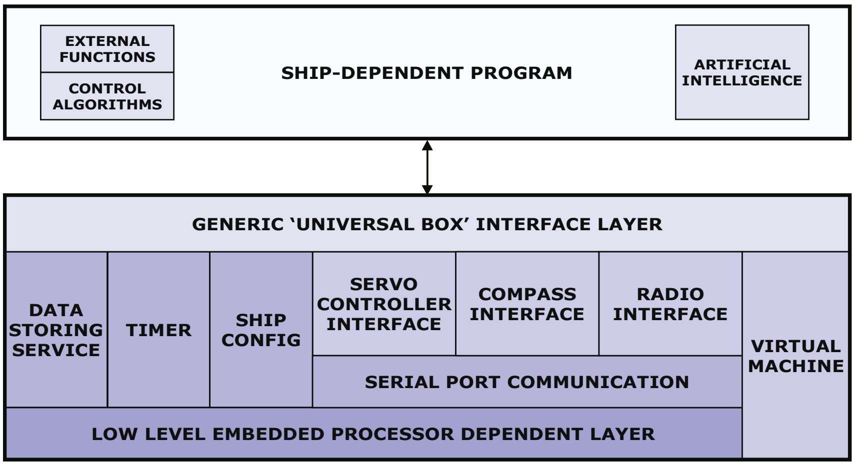


Fig. 21. On-board software architecture

As depicted in the figure, we have a main process that uses lower modules, organized as a software library. This library offers high level routines to handle the ship, and isolates coordination, decision and control from low level tasks, like timers, virtual machine execution and serial port communication. These tasks work in a parallel way, simulating real multi-threading with techniques like non-blocking calls and active polling. With this design, every ship uses the same library, and we only have to program the particular behavior of each one.

The ships offer public, particular functions that we can call in ship-language programs to test different solutions and algorithms in experimental environments, programming them remotely, as explained later.

There are some things about architecture that should be noted. We have a layered design, in which each module interacts only vertically. A software module never calls or receives information from a module of the same level in the architecture: it only uses the services provided by the inferior layers, and provides services to the higher level modules. This way we keep the software uncoupled, and gain power with reusability and encapsulation. Only the main module of each ship uses all the components of the system, calling the functions offered in the library, maintaining a centralized process based on services. This could seem excessive for an embedded application, heavily oriented to hardware control, but the experiments and continuous tests have proven that this way the research is faster and easier.

Perhaps the most important feature that has been implemented is our ship-language, and the ship-language interpreter. With this ship-language the ship behavior (motions, maneuvers, exchange of messages and orders) can be defined. Our intention is to research and create a language that allows defining complete and complex enough experiments, from the beginning to the end of the tests. This simple but powerful language somehow resembles a simple scripting language. It's intended to be very clear and small; not only for a computer science expert but for any technical discipline expert who knows the principles of high level programming.

We have developed a compiler that translates high level code (ship-code) to stack language code that the ships can run. This translation produces faster, smaller and simpler to execute code, which is fundamental for our processor. Once written and compiled in the central computer, the code is packed and sent to the ship, which can then execute it. Fig. 22 shows a diagram about the *ship-language interpreter*:

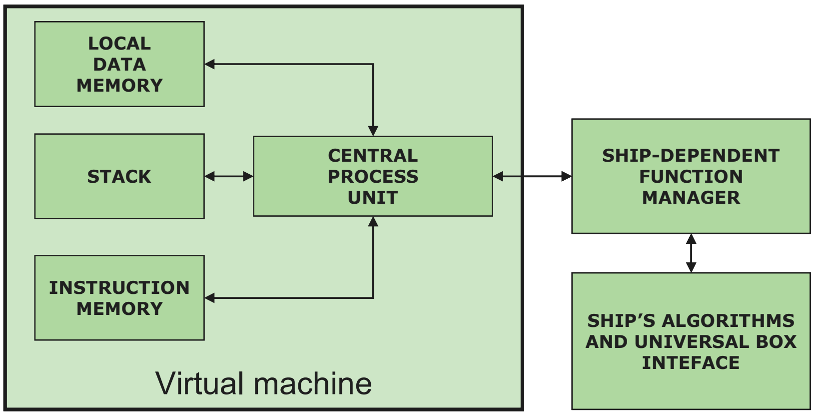


Fig. 22. 'Ship-language' interpreter

The previous figure depicts how does the stack language code executor (in a global view) works. There is a low level machine (the virtual machine itself) that handles low level control orders coded in stack language: conditionals, jumps, arithmetic-logical operations and local memory transactions. When the virtual machine executes an order that is not on his repertory, it "gives" it to the function manager, which is located on the main module and runs very high level orders like "go to", "send message" or "set speed". The important thing of this manager is that it enables communication between the virtual machine and the rest of the system: shared data, ship control, communication, timer, and others. Each ship exports all those functions that it can execute, and there are many of them that all the ships can handle. But it's important to note that each ship can execute the same function in a different way (the ships are not equal), and that some ships will have some unique functions.

For example, the language can be used to program how to capture a buoy. We have two ships, 1 and 2. Ship 1, the leader, goes to the buoy, and waits for ship 2 to arrive. Then, they both return to the base with the buoy. The ship-code of ship 1 could be like the following (figure 23):

```
# Ship 1 code

# Environment variables
xBase = 10
yBase = 20

# Go to automatic mode
automaticMode()

# The ship must be 3 meters away from ship 1
keepDistance(2, 3)

# We are the group leader
sendLeader(1)

# Set speed
speed(4)

# First we go to the buoy, which can
# be moving
while not reachBuoy()
    xBuoy, yBuoy = getBuoyPosition()
    setDestiny(xBuoy, yBuoy)
end

# Face south
orientation(270)

# We wait for ship 1, then we continue
stop()
waitForArriveSignal(2)
speed(4)

# We signal ship 2, and it will know
# that it should continue with its program
sendContinueSignal(2)

# On board algorithm to capture the buoy.
# When the ship captures the buoy, with
# the help of its team, the program
# can continue
captureBuoy()

# If everything went ok, we return to base. # Else, we
send an error message.
if buoyCaptured()
    signal('Returning to base')
    setDestiny(xBase, yBase);
else
    signalError('Can't capture buoy')
end
```

Fig. 23. Example of 'ship-language'

7.1 Communication Between Ships

Ships' communication protocol is based on many common and simple techniques of net topologies. Communication is one of the most important matters of this research, that's why we have put much effort on correctness and effectiveness.

Many classic net systems usually keep track of errors with an acknowledge packet (ACK), which, in our case, saturates the communication: two packets for one unit of information. Unlike that, we use a 'WHAT' message, which indicates that something has not been well recognized. When a ship receives a message and finds that it has no sense, it sends this type of signal to the sender ship, which interprets it as a petition of resending some packet.

Our radio link is very reliable (in many experiments we have had no errors at all), and the radio-modems we use control media access and packet collision, so we have only to take care of signal transmission errors, that's why we have applied these ideas.

7.2 Cooperation

There are several conceptual alternatives that can be considered to implement cooperation protocols. The idea we adopted is to use a "shared memory". All ships have a copy of the shared memory, which keeps track of the status of each ship, and enables every unit to suppose what the other ships are doing. This memory is periodically refreshed with info-packets from the units of the team, and can be considered as the state memory of the group. We use typical cache memory techniques to maintain coherency and consistency inside the team of ships, because every ship must know the newer values of every variable. Any unit in the group can set any variable of any ship, because probably the external unit (other ship, or a human managing the base), knows better the value of some variable.

With this shared memory we share not only status data like position and orientation, but also high level knowledge like intentions and believes about the environment, which are crucial for cooperative behavior, intelligent behavior and decision making. With this approach, for example, a ship can share its objective, and other ship, without an explicit order, can help the first ship.

Fig. 24 shows a diagram with the shared memory concept:

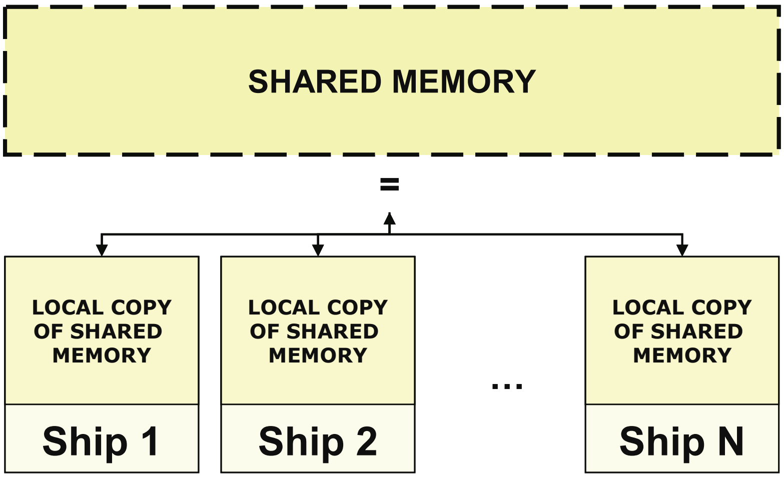


Fig. 24. Shared memory concept

With all these concepts we can build an agent cooperation system, based on common information and explicit orders. According to an object-oriented perspective, each ship takes own decisions. We know every ship can run a set of defined orders, but the same order is obeyed in a different way by each different ship. A patrol ship has not the same dynamic characteristics as a tanker, and they should not behave the same way.

8 Experiments

Experiments are done using a 150m x 30m basin. This is one of the basins in a towing tank facility (Canal de Experiencias Hidrodinámicas de El Pardo, CEHIPAR) which is near our university. The basin has a wavemaker.

Our initial experiments focused on the individual behavior of the autonomous ship. Several maneuvering experiments have been done, to obtain simple dynamic models of the ship, and to develop and test the motion control.

Initial cooperation scenarios have been defined for experimental study. Experiments on two formation cases have been already done.

An experimental support system has been developed. This system consists in a portable computer with a digital radio link that can exchange information with the ships from ashore. The behavior of the ships can be programmed from distance, using this support system and programs written in ship-language, so we can give orders and watch the results. We can see the progress of the experiments and the internal status of the ships with a 3D graphical interface, and get the data collected during the experiments, which can we use for reproduction of an experiment, analysis and behaviour interpretation. We show a snapshot of this tool in Fig. 25.

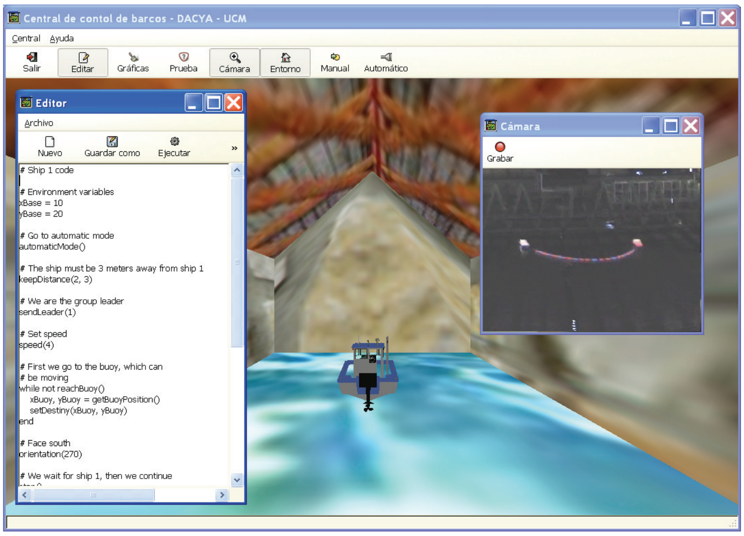


Fig. 25. Control application

Fig. 26 shows a diagram with the experimental support system:

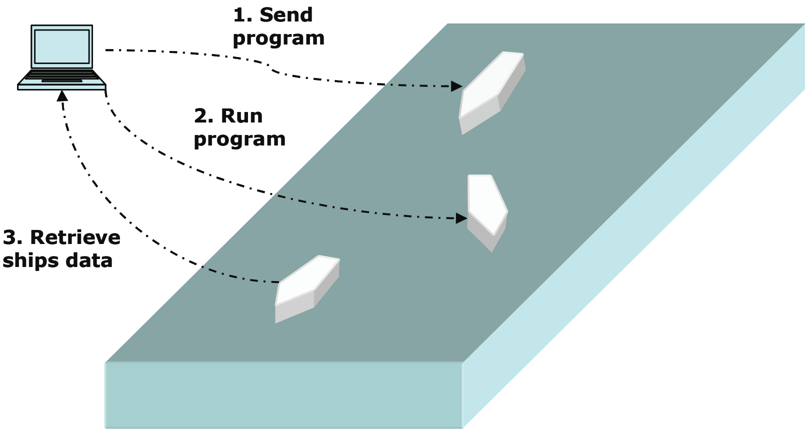


Fig. 26. The experimental support system

We can see, in Fig. 27, a photograph of an experiment, with two ships towing a buoy towards the base:

Fig. 27. Two autonomous ships towing a buoy

9. Conclusion

This chapter presented an overview of a research on ship automatic cooperation. The research includes several aspects, such the study in simulation of the dynamics of two ships towing a boom, the establishment of a genetic control planning procedure to generate good trajectories for the ships in any marine operation scenario, and finally scaled autonomous ships for cooperative robotic studies in marine scenarios. The main ideas for cooperation implementation have been introduced..

In the future the experimental scenarios will become more complex, and, as complexity grows, perhaps the “ship-language” should be empowered to cover more requirements of the experiments, allowing it to express higher level orders that ships should understand, so that we need no sequential programs to express operations.

Acknowledgments

The authors want to thank the support of the CICYT Spanish Committee (project DPI2003-09745-C04-02) and the collaboration of the CEHIPAR staff.

References

Aguiar, P.A. and A.M. Pascoal .2002. Practical stabilization of the extended nonholonomic double integrator, In Proceedings 10th. Mediterranean Conf. on Control and Automation, Lisbon, Portugal,.

Alvarez, A., A. Caiti and R. Onken .2004. Evolutionary path planning for autonomous underwater vehicles in variable ocean, IEEE Trans. Oceanic Eng., vol. 29, n.2, April, pp. 418-429

Baillieul, J. and A. Suri .2003. Information patterns and hedging Brockket´s theorem in controlling vehicle formations, In Proceedings 42nd. IEEE Conf. Decision and Control, Maui, Hawaii, pp. 556-563.

Balch, T. and R.C. Arkin .1998. Behavior-based formation control for multi-robot teams, IEEE Trans. Robotics and Automation, v. 14, n. 6, pp. 926-939.

Betts, J.T. .1998. Survey of numerical methods for trajectory optimization, J. Guidance, Control, and Dynamics, v.21, n.2, pp. 193-207.

Billard, E.A. .2004. Patterns of agent interaction scenarios as use case maps, IEEE Trans. System, Man and Cybernetics*,* PartB, vol. 34, n4, August, pp. 1933-1939.

Bloch, A. .2003.Nonholonomic Mechanics and Control, Springer Verlag.

Brockett, R.W. .1990. Some mathematical aspects of robotics, In Proceedings of Symposia in Applied Mathematics, vol.41, American Mathematical Society, Providence.

Cao, Y.U., A. S. Fukunaga and A.B. Kahng .1997. Cooperative mobile robotics: antecedents and directions, Autonomous Robots, v.4, pp. 7-27.

Esteban, S., B. Andres-Toro, E. Besada-Portas, J.M. Giron-Sierra and J.M. De la Cruz .2002. Multiobjective control of flaps and T-foil in high speed ships, In Proceedings IFAC 2002 World Congress, Barcelona.

Farinelli, A., Iocchi, L. and Nardi, D., Multirobot systems: A classification focused on coordination. IEEE Transactions On Systems, Man, and Cybernetics- Part B: Cybernetics, Vol. 34 no.5, October, 2004.

Fossen, T.I. .2002. *Marine Control Systems,* Marine Cybernetics AS, Trondheim..

Giron-Sierra, J.M., S. Esteban, J. Recas, B. De Andres, J.M. De la Cruz, J.M. Riola, J. Aranda and F. Velasco .2005. Overview of a research on actuators control for better seakeeping in fast ships. In Proceedings IFAC World Congress 2005, Prague.

Giron-Sierra, J. Jiménez, A. Dominguez, J.M. Riola,, J.M. De la Cruz and B. De Andres .2005. A cooperation scenario in the marine environment: first outlook.. In Proceedings IFAC World Congress 2005, Prague.

Goldberg, D.E. .1989. Genetic Algorithms in Search, Optimization and Machine Learning, Addison-Wesley.

Ikeda T, Mita T. Masaki Y. .2005. Formation control of underwater autonomous vehicles. In: Proc. 16th IFAC World Congress. Praga.

Johansen, V., R. Skjetne and A.J. Sorensen .2003. Maneuvering of towed interconnected marine systems, In Proceedings IFAC MCMC 2003, Gerona, Spain, pp. 292-297.

Kumar V., Leonard N., Morse A. S .Eds.) .2005. Cooperative Control, LNCIS 309. Springer-Verlag, Berlin.

Kwiesielewicz, M., W. Piotrowski , R. Smierzchalski and R. Sutton .2000. AUV path planning for navigational constraints by evolutionary computation, In Proceedings IFAC MCMC 2000, pp. 323-326.

Kyrkjebo, E. and K.Y. Pettersen .2003. Ship replenishment using synchronization control, In Proceedings IFAC MCMC 2003, Gerona, Spain, pp. 286-291.

Kyrkjebo, E., M. Wondergem, K.Y. Pettersen and H. Nijmeijer .2004. Experimental results on synchronization control of ship rendezvous operations, In Proceedings IFAC CAMS 2004, Ancona, Italy, pp. 453-458.

Lapierre, L., D. Soentanto and A. Pascoal .2003. Coordinated motion control of marine robots .2003), In Proceedings IFAC MCMC2003, Gerona, Spain, pp. 250-255.

LaValle, S.M. .2006. Planning Algorithms, Cambridge Univ. Press .to appear. Available at http://msl.cs.uiuc.edu/planning/

Liu, J. and J. Wu .2001. Multi-Agent Robotic Systems, CRC Press.

Lloyd, A.R.J.M. .1998. Seakeeping: Ship Behaviour in Rough Weather, A.R.M.J. Lloyd, Gosport, Hampshire, U.K.

Michalewicz, Z. .1999. Genetic Algorithms + Data Structures = Evolution Program, Springer-Verlag.

Miettinen, K.M. .1999.Nonlinear Multiobjective Optimization, Kluwer.

Morishita, H.M., E.A. Tannuri and T.T. Bravin .2004. Methodology for dynamic analysis of offloading operations, In Proceedings IFAC CAMS 2004, Ancona, Italy, pp. 459-464.

Parker, L., Current state of the art in distributed robot systems. Distributed Autonomous Robotic Systems 4. .Eds.; Parker, L., Bekey, G. and Barhen, J.) Springer, 2000, pp. 3-12.
Skjetne, R., I.F. Ihle and T.I. Fossen .2003. Formation control by synchronizing multiple maneuvering systems, In Proceedings IFAC MCMC 2003, Gerona, Spain, pp. 280-285.
Soetanto, D., L. Lapierre and A. Pascoal .2003. Coordinated motion control of marine robots .2003), In Proceedings IFAC MCMC2003, Gerona, Spain, pp. 250-255.
Stilwell, D.J. and B.E. Bishop .2000. Platoons of underwater vehicles, IEEE Cotrol Systems Magazine, vol. 20, December, pp. 45-52.
Tan, C.S., R. Sutton and J. Chudley .2004. An incremental stochastic motion planning technique for autonomous underwater vehicles, In Proceedings IFAC CAMS 20004, Ancona, Italy, pp. 483-488.
Tanner, G.T., J. Pappas and V. Kumar .2004. Leader-to-formation stability, IEEE Trans. Robotics and Automation, vol. 20, n.3, June, pp. 443-455.
Weiss, G. .1999. Multiagent Systems. MIT Press.

INDEX

ISBN-13: 978-84-611-3915-6
ISBN-10: 84-611-3915-1

AUTOMAR Thematic Network was aiming to foster Spanish research and innovation activity in the maritime industrial sector in order to strengthen their role in Europe. During the six years of the AUTOMAR Network many activities have been carried out, and all the partners have actively contributed. This book, prepared to be presented at the last AUTOMAR meeting, collects in its 14 chapters excellent contributions that show the level of scientific knowledge in this field reached by the Spanish RTD, both at the Universities and at Research Institutions

This book is intended to disseminate the knowledge and applications of the Spanish research groups working on robotics and automation in the maritime industries